NASA Contractor Report 194452

Data Generation for Destination Directed Packet Switch

D.J. Shyy and Tom Inukai
Comsat Laboratories
Clarksburg, Maryland

April 1996

Prepared for
Lewis Research Center
Under Contract NAS3–25933

National Aeronautics and
Space Administration

TECHNICAL SUPPORT FOR DIGITAL SYSTEM TECHNOLOGY DEVELOPMENT

Task Order No.4

Final Report

DATA GENERATION FOR DESTINATION DIRECTED PACKET SWITCH

Submitted to

National Aeronautics and Space Administration
Lewis Research Center
2100 Brookpark Road
Cleveland, Ohio 44135

Contract No. NAS3-25933

July 23, 1993

COMSAT LABORATORIES
22300 COMSAT DRIVE, CLARKSBURG, MARYLAND 20871

Table of Contents

Table of Contents (cont'd)

List of Illustrations

List of Illustrations (cont'd)

List of Illustrations (cont'd)

List of Tables

List of Tables (cont'd)

Section 1
Introduction

A destination directed packet switch (DDPS) for future advanced satellite communications employs a simple routing concept based on the routing information contained in individual packet headers. Several institutions, including the Lewis Research Center of National Aeronautics and Space Administration, are currently developing a proof-of-concept DDPS to exploit its capabilities for providing future multimedia user services. One critical issue in DDPS development is a procedure of testing such a system under various simulated traffic environments. Although some commercial test equipment are currently available for testing general purpose switching systems and Asynchronous Transfer Mode (ATM) switches, they are often constrained by simple traffic models, fixed packet formats, and no dynamic control of traffic sources. The purpose of this study is to investigate a design concept of traffic generator to be used for tesing an on-board DDPS in a testbed environment.

The major functions of the DDPS testbed include traffic generation, verification of switching/traffic management protocols, and performance measurements. To test a high-speed DDPS, the traffic generators must be capable of creating an actual traffic load with different characteristics in real time. These traffic generators can generate traffic patterns, which emulate the behaviors of real traffic sources, for testing the switching function of the DDPS and incorporate a traffic source response procedure to congestion for testing congestion control algorithms. Switching and congestion control are considered to be two of the most critical design elements for an on-board DDPS [1-1]. In addition, the traffic generators can be used to evaluate switch performance, properly size the on-board buffer, evaluate the performance of traffic management functions such as admission control, priority control, and capacity allocation for sources with different characteristics, and measure the traffic performance through the satellite network.

In this study, two types of traffic sources are considered: voice and data. Various queueing modeling techniques are investigated to characterize a single source and multiplexed traffic sources, which include a constant bit rate process, Poisson process, Geometric process, batch Poisson process, Hyperexponential process, on-off process, and modulated Markov Poisson process. The stochastic process parameters for a single voice channel are well documented; however, it is not easy to choose the parameters for multiplexed voice and data traffic. To aid the user to select the stochastic process parameters, alternative procedures of fitting the measurement results from real traffic sources to a queueing model are also addressed.

The report includes alternate approaches of implementing a traffic generator. The hardware/software requirements for different approaches are compared. The necessary modification of the traffic generator to incorporate a traffic source response procedure to congestion is addressed. Also presented are the configuration for the testbed and the functionalities of its components, including traffic generators, a network simulator, a network controller, and packet receivers.

This report is organized as follows.

Section 2 describes the satellite network requirements which include the reference network architecture and packet formats and includes suggestions for the development of a refined archtiectrue. It also summarizes the key functional requirements and features of the traffic generator.

Section 3 presents a number of queueing models to characterize data and voice traffic patterns. Alternate procedures of fitting a measured traffic pattern from real data traffic sources to a queueing model are also described.

Section 4 provides a trffic analysis for the DDPS based on traffic generator prameters. The analysis procedure is used to select traffic generator parameters to induce switch congestion in a controlled manner. An overall control procedure for DDPS testing based on analysis and synthesis techniques is described.

Section 5 discusses different approaches of implementing a traffic generator. The hardware/software requirements for alternate approaches are compared. The configuration of the testbed and its components, which include traffic generators, a network simulator, a network controller, and packet receivers, are described.

Section 2

Requirements Analysis

This section presents a description of the underlying network architecture for the data generation study, general requirements for the traffic simulator, and technical considerations. Analyses of key system requirements in terms of their implications and impacts on simulator design are also addressed.

2.1 Underlying Network Architecture

The work performed in this study assumes the network architecture described in Reference [1-1]. The satellite network under consideration operates at the 30/20-GHz frequency band and provides flexible, low-cost mesh VSAT services to users located in the continental United States (CONUS). The satellite antenna coverage consists of eight fixed uplink beams and eight hopping downlink beams, where each downlink beam visits eight dwell locations. In addition, the system may include intersatellite links (ISLs) to support direct connection to other regions of the world. An on-board baseband processor (OBP) provides connectivity among uplink and downlink beams as well as ISL transmission. Figure 2-1 depicts the system concept.

The network provides voice, data, facsimile, datagram, teleconferencing, and video communications services. To support these services, the system employs multi-frequency time-division-multiple-access (MF-TDMA) for uplink transmission and burst TDM for downlink. Each uplink beam contains thirty-two 2.048-Mbit/s TDMA carriers (an aggregate transmission rate of 67 Mbit/s) with a frame period of 32 ms. Downlink transmission is single-carrier TDM with the same capacity and frame period as the uplink beam. The actual transmission rates for uplink and downlink will be higher than the rates given above due to FEC coding and overhead.

On-board traffic routing is performed by a destination directed packet switch (DDPS), which routes uplink data packets to the proper downlink beams according to the routing information contained in the packet headers. The packet consists of 2048 bits, corresponding to a 64 kbit/s capacity for a periodic transmission of once per frame (i.e., 2048 bits/32 ms = 64 kbit/s). To minimize the amount of on-board storage, each packet is transmitted in sixteen subpackets, where a subpacket consists of 128 bits. The first subpacket is a packet header, and the other fifteen subpackets contain user information. This results in a user information rate of 60 kbit/s for a periodic transmission. The frame structure and packet format are also shown in Figure 2-1. Since the proposed system employs bit synchronous burst timing, no burst preamble is needed. (Note that the actual system may require a minimum of one symbol guard time between bursts to accommodate a non-ideal burst transient response. Also, a larger guard time must be allocated for initial transmit timing acquisition.)

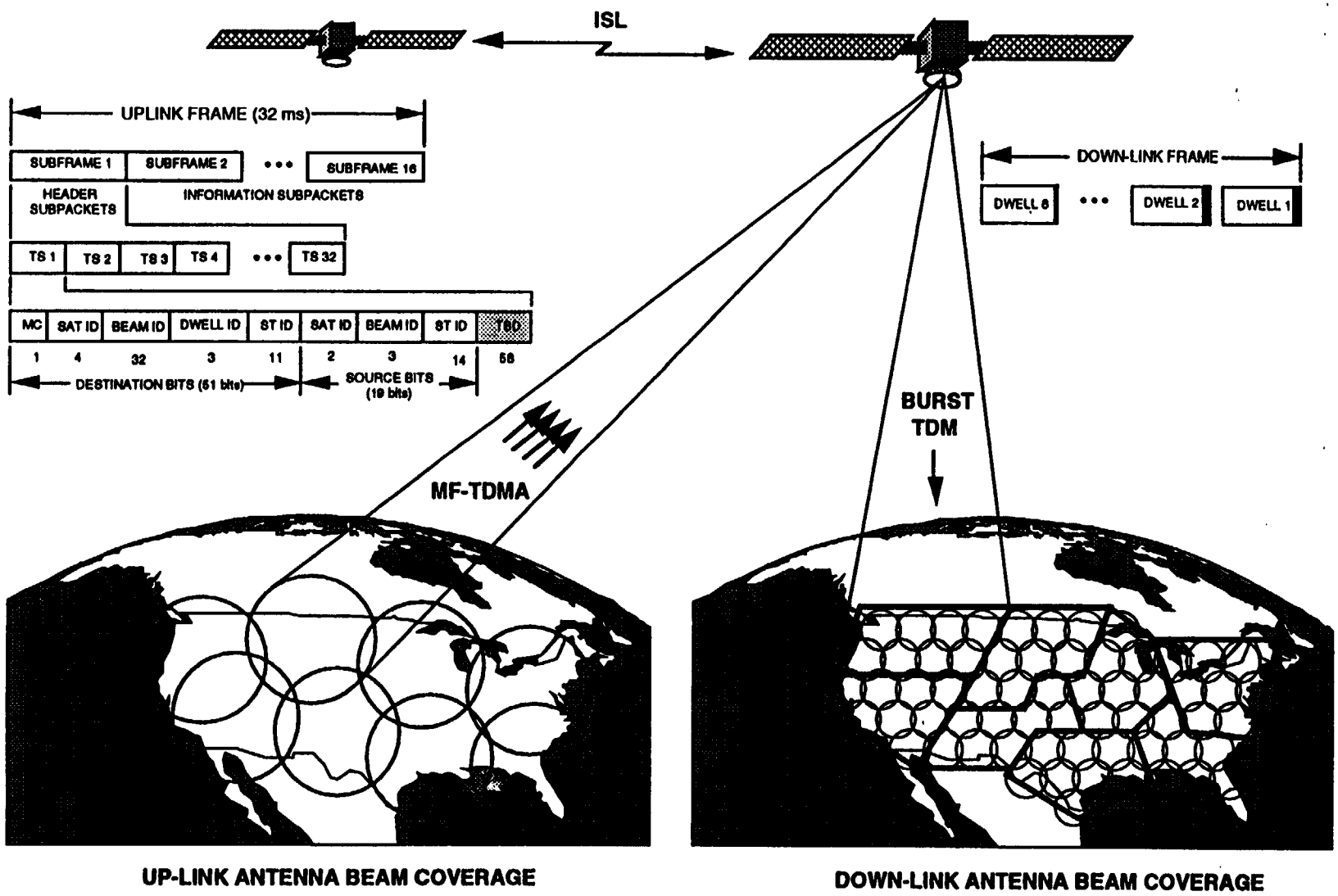

Figure 2-1. Network Architecture

The on-board switch performs the following functions. During Subframe 1, it receives up to 32 header subpackets from each TDMA carrier and stores routing information in a control memory. The subpackets in Subframe 1 and all subsequent subpackets in Subframes 2 through 16 are routed to the proper downlink TDM buffers according to the stored routing information. In the next frame, the control memory is refreshed with the new routing information contained in the header subpackets of the new Subframe 1. Downlink packet transmission is synchronized with subframe timing such that header subpackets are transmitted in the first downlink subframe followed by 15 information subpackets in the next subframes. To maintain subframe synchronization, subpackets are queued in a group of 16 subpackets constituting the origital packet and transmitted to a dwell area in the next 16 subframes. When congestion occurs, subpackets are discarded in a group of 16 subpackets for a queued packet. On the other hand, deletion of new subpackets (not queued subpackets) will occur on the same subpacket slot number over 16 consecutive subframes to avoid disruption to the order of subpacket sequence.

The packet header structure is also depicted in Figure 2-1. A few comments are provided on the proposed structure in the following. The satellite identification field (4 bits) in the destination bits is somewhat redundant, since the position of the beam identification field uniquely identifies the destination satellite. The header address provides terminal-to-terminal connection but not information of discriminating different circuit connections between the terminals. An additional field, such as a virtual channel number (VCN), may be included in the header to uniquely distinguish different circuit connections. Special VCNs may be reserved for signaling purposes. Alternately, terminal port numbers can be used but will not be as flexible as a VCN, since one physical port may be connected to a number of data terminal equipment (DTE) via a local area network (LAN). The use of VCN may also eliminate destination and source terminal identification fields. A further study is recommended on the design of an optimal header structure.

The DDPS system provides multicast/broadcast connection to different downlink beams. However, to minimize a packet overhead, a multicast packet is broadcast to all dwell areas of a destination beam. This gives rise a potential problem for multicasting to a few dwell areas of multiple beams. An extreme case occurs for multicasting to a single dwell area of each downlink beam. In this case, a single uplink packet will occupy sixty-four downlink time slots, resulting in a 700-percent wasted downlink capacity. The problem may be mitigated by a simple algorithm to combine point-to-point and multicast transmission based on the amount of multicast traffic. Alternately, an on-board multicast VCN-beam/dwell translation table may be used to alleviate the problem.

2.2 Requirements for Traffic Simulators

The main object of this study is to develop a procedure of generating realistic traffic for the purpose of testing a DDPS, in particular the performance of the DDPS for handling contention and congestion situations. A possible test configuration for the DDPS is shown in Figure 2-2. The size of the on-board switch is 8 x 8, corresponding to eight uplink and eight downlink beams. Data packets routed to a switch output port are

sorted for downlink transmission according to eight destination dwell areas. Thus, the DDPS provides an 8 x 64 beam/dwell area switching function.

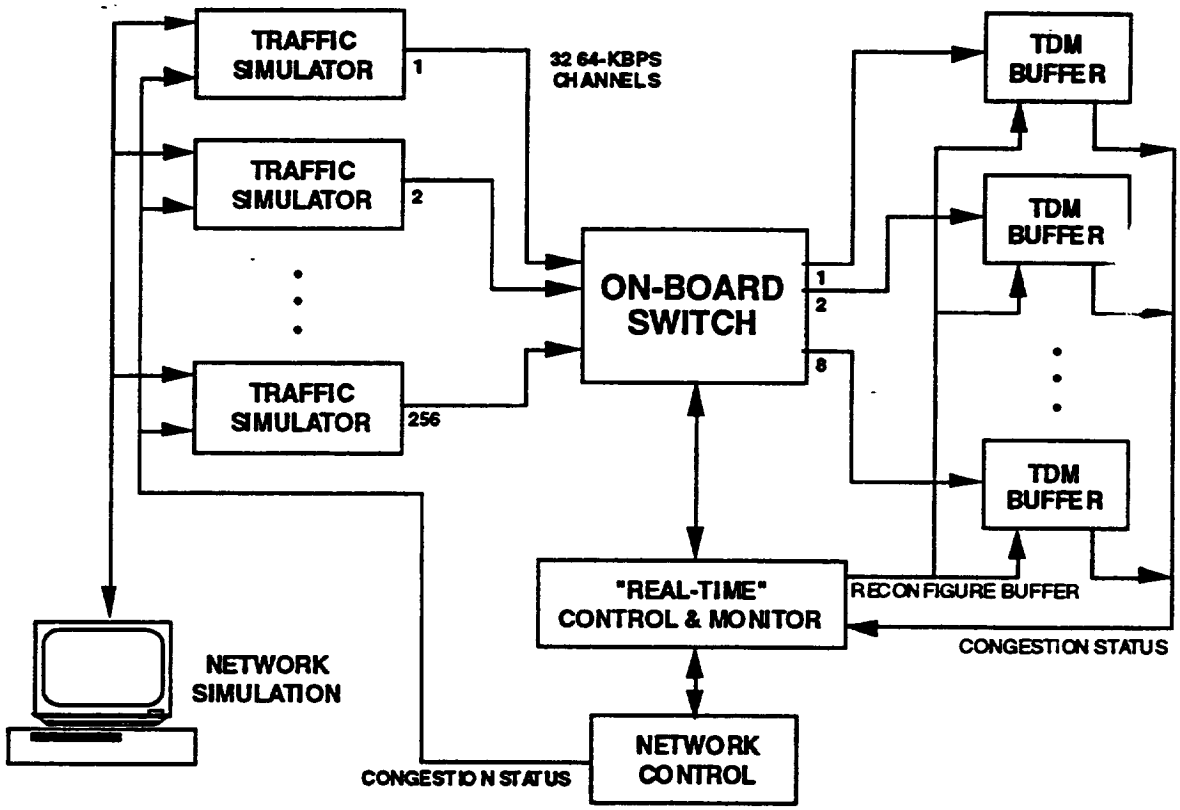

Figure 2-2. DDPS Test Configuration

The traffic simulators generate data packets that represent realistic communications traffic scenarios for the mesh VSAT network. The total traffic capacity of the DDPS is 8192 packets per frame (PPF) which is equivalent to 8 beams x 32 carriers/beam x 2.048 Mbit/s/carrier = 524.288 Mbit/s. The number of traffic simulators ranges from eight to 256, each generating 1024 PPF or as little as 32 PPF. In the test configuration using eight traffic simulators, each simulator emulates aggregate traffic from one uplink beam at the transmission rate of 65.536 Mbit/s. In the 256-traffic simulator configuration, a simulator emulates traffic of a single uplink TDMA carrier at 2.048 Mbit/s. In this report, the first test configuration is considered, since it simplifies simulator implementation as well as operation of the testbed. The simulator equipment may be compactly packaged on a single chassis with eight simulator boards.

The traffic simulator must be capable of generating various traffic patterns to test the performance of the DDPS hardware under normal and stressed traffic environments. Specifically, the following features must be incorporated in the traffic simulator:

a. The packet headers must be representative of "real" communications traffic and be continuously varying in order to emulate "real" traffic scenarios.

b. A mechanism for identifying input and output packets must be developed in order to trace and verify the data flow of a particular circuit connection.

c. Upon reception of a congestion notification from the switch, the packet generator must implement congestion control by modifying the traffic flow to the particular point of congestion.

d. A mechanism for forcing congestion to occur at any dwell must be identified.

e. A mechanism for forcing multicasting and broadcasting must be identified.

For Item a, Section 3 of this report presents various candidate traffic models which have been used in the past to model realistic traffic flows and recommends a particular model for potential implementation.

Item b requires the use of a VCN (or other equivalent techniques) and a packet sequence number (PSN) to uniquely distinguish different circuit connections and to trace a sequential packet flow or to identify the packets discarded by the switch during congestion.

Item c may be implemented in conjunction with the network control function. The network controller, depicted in Figure 2-2, continuously monitors on-board queue status in the downlink TDM buffer, makes decisions on when to perform congestion control based on its internal algorithm, and implements satellite link propagation delay for congestion control messages to be delivered to the traffic simulators. The congestion control message consists of the following:

- Identification of the downlink beam or dwell area for congestion control

- Traffic volume adjustment factor α ($\alpha > 1$ to increase traffic, $\alpha < 1$ to decrease traffic)

The traffic simulator simply modifies its traffic flow based on the congestion control message supplied by the network controller. In the operational system, the above congestion control functions performed by the network controller may reside within user terminals. However, the proposed technique allows centralized control of the test configuration and provides flexibility of changing a control mechanism as desired.

Item d may be regarded as a traffic synthesis problem, i.e., to set up traffic simulator parameters such that congestion occurs at a selected beam or dwell area in a controlled manner. Possibly, the best measure to quantify the degree of congestion is a packet loss ratio (PLR) which is directly related to the amount of traffic flow. Section 4 describes a procedure of generating a set of traffic simulator parameters to induce congestion for a given PLR. The types of congestion depend on the structure of downlink TDM buffers. If a common buffer is used for all dwell area traffic within a single beam and downlink

time slots are dynamically allocated, then beam congestion occurs. If separate buffers are allocated for different dwell areas, congestion occurs only in certain dwell areas. Implication of these alternative buffer designs on congestion control is further addressed in a latter section.

Item e can be satisfied with incorporation of a multicast indication field in the packet header.

Section 3

Traffic Source Queueing Models

Packets arriving at the on-board DDPS are a superposition of packets of different traffic types (such as data and voice) from ground terminals. The multiplexed, multimedia traffic often exhibits complex, bursty, and high-variation patterns. The nature of some data traffic types such as computer communication traffic tends to be bursty. Traffic burstiness can be characterized by the interarrival time of successive packets within a burst. It has been recognized that bursty traffic degrades the switch performance; as a result, the switch buffer must be sized large enough by considering the traffic burstiness. The bursty nature of an uplink multiplexed packet stream may result in short term network congestion; consequently, the amount of capacity allocated for a bursty connection must be larger than that for a less bursty connection. Accurate traffic source models are crucial for network engineering (e.g. admission control) and performance evaluation (e.g. packet loss ratio of the switch). In this section, different traffic source queueing models for data, voice and multiplexed streams are investigated.

To completely analyze the performance of a DDPS, three layers of traffic flows and their statistical behaviors and interactions need to be modeled. These three layers are calls, bursts and packets, and they manifest themselves in different time scales. In this study, only two layers are considered: calls and packets [3-4]. The number of calls established for each user terminal is part of admission control or traffic management. For data traffic, it is assumed that the number of connections from a source is fixed. The data call setup and release traffic model will not be discussed; only packets layers are discussed. For voice traffic, both layers are addressed. It is assumed that no call blocking will be experienced by voice calls. Voice call arrivals and voice call duration follow certain distributions.

This section investigates various queueing models to characterize the packet layer traffic pattern, i.e., to generate the packet arrival time for different traffic sources in the traffic generator. It has been recognized that to build a complete analytical model or simulation model for different traffic types is extremely difficult [3-1]. Complex source queueing models are simplified in this study for analytical tractability and ease of implementation. Typical parameter values of the queueing models used in the literature are listed if available. Some traffic models are capable of capturing the burstiness, and others are not. The burstiness and variability of these traffic models can be characterized by several measures: the coefficient of variance and the correlation between packets such as index of dispersion for intervals or counts [3-1][3-2][3-3].

Among the queueing models presented in this section, the two-state Markov-Modulated Poisson process (MMPP) is recommended for modeling the packet layer traffic pattern. The two-state MMPP is capable of capturing traffic burstiness and correlation and is simple for analysis. The two-state MMPP is also the most generalized and widely used model for bursty traffic and a superposition of packetized voice and data traffic. The two-state MMPP was used to fit the measured data from a workstation and the fitting

result was encouraging [3-3]. Procedures of determining its associated parameters using measurement data are also discussed.

3.1 Notations

This subsection introduces the notations and terminologies used in this report.

From the traffic source point of view, the packet interarrival time is the interval between the beginning of the transmission of a packet and the beginning of the transmission of the previous packet. Clearly, the interarrival time ≥ packet transmission (slot) time for a slotted system. From the packet receiver point of view, the packet interarrival time is the interval between the receiving time of a packet and the receiving time of the previous packet. The latter definition is adopted in this study.

Let C_n denote the nth packet to arrive at a queueing system, and τ_n denote the arrival time for C_n.

$$t_n = \tau_n - \tau_{n-1} = \text{interarrival time between } C_n \text{ and } C_{n-1}.$$

Define a random variable t as the interarrival time. Associated with each is a probability distribution function (PDF), i.e., $A(t_0) = P(t \le t_0)$ and a related probability density function a(t). The first two moments associated with the random variable t are E[t] and $E[t^2]$. The first moment is also referred as mean or expected value.

$$E[t] = \int_0^\infty t\, a(t)\, dt, \tag{3-1}$$

$$E[t^2] = \int_0^\infty t^2\, a(t)\, dt, \tag{3-2}$$

The variance (or dispersion) of random variable t is

$$\text{var}[t] = (\sigma_t)^2 = E[t^2] - (E[t])^2, \tag{3-3}$$

and the positive square root of the variance is called the standard deviation (σ_t).

The renewal process is described below. Let {N(t), t ≥ 0} corresponds to a series of points on the time interval from 0 to infinity. {N(t)} is a renewal process if the interarrival times between successive points are independently and identically distributed random variables.

To properly characterize the variability and burstiness of different traffic models, three measures can be used. The first is to use the coefficient of variance (CV), defined as the ratio of standard deviation to the mean value [3-2]:

$$\text{Coefficient of Variance: } CV = \frac{(\sigma_t)}{E[t]} \qquad (3\text{-}4)$$

CV has a value 0 for the deterministic process and 1 for the Poisson Process [3-1]. CV is useful for observing the variation of individual packet arrival time and comparing the standard traffic sources, such as deterministic (CV = 0) and Poisson (CV = 1). Most researchers characterize a traffic source bursty when its CV is larger than 1 (see Figure 3-1). There is no formal definition of the packet arrival process when CV is less than 1. Although CV can be used to observe the burstiness of the packet arrival process, CV can not measure the correlation between packets.

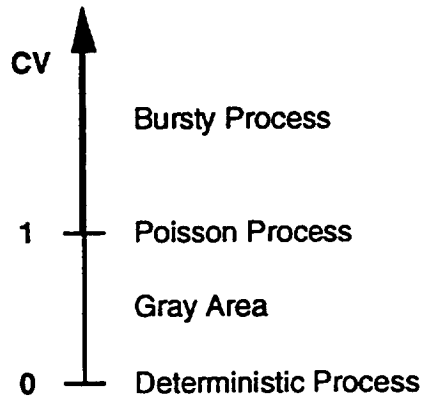

Figure 3-1. Physical Meaning of Coefficient of Variation

The second technique to characterize burstiness is the index of dispersion for intervals [3-3]. The index of dispersion for intervals (IDI) is a dimensionless quantity and is a measure of the variations of the sum of packet interarrival times. Consider n random variables (T_1, ..., T_n), where a random variable represents the interarrival time between two successive packets. The n-th order IDI is defined as

$$\text{Index of Dispersion for Intervals (IDI): } J_n = \frac{\text{var}[T_1 + ... + T_n]}{n\,(E[T])^2} \qquad (3\text{-}5)$$

Let c(h) be the autocovariance function.

$c(h) = \text{cov}(T_j, T_{j+h}) = E[\,(T_j - E[T])\,(T_{j+h} - E[T])\,] = E[T_j, T_{j+h}] - (E[T])^2$, where h is an integer.

$$\text{var}[T_1 + ... + T_n] = n\,\text{var}[T] + 2\sum_{j=1}^{n-1}\sum_{h=1}^{j} c(h)$$

Note if x and y are uncorrelated random variables, E[xy] = E[x] E[y].

If x and y are statistically independent random variables, $f_{xy}(x,y) = f_x(x)\, f_y(y)$ and E[xy] = E[x] E[y], where f() is the probability density function (pdf).

Clearly, if all n interarrival times are independent, then the variance of the n interarrival times is equal to n times the variance of the original interarrival time. If they are correlated, the variance will be lager than n times the variance of the original process. The first order of IDI J_1 is CV^2.

$J_n = 0$ for all n for a deterministic process, $J_n = 1$ for all n for the Poisson process, and J_n = constant for all n for the renewal process.

Since the correlation between T_j and T_{j+h} becomes smaller when $|h|$ becomes larger, J_n will reach a limit (constant) when n is sufficiently large.

Remember that J_n is to compute the variance of groups of interarrival times, where the group size is n. If the variance of the groups of interarrival times is larger than n times the variance of the original process, it suggests that clustering of small interarrival times and large interarrival times occurs in different groups. This situation is largely due to computer communications protocols such as fragmentation and client/server paradigm [3-3]. An example of packet interarrival times is illustrated in Figure 3-2. These interarrival times are used to compute J_1 and J_2.

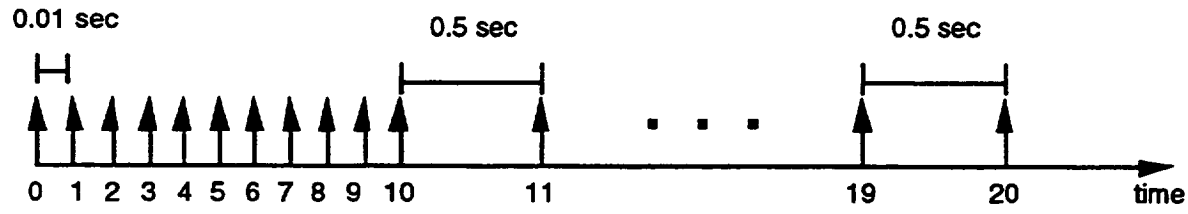

Figure 3-2. Example Used to Compute IDI

First, we compute J_1:

$E[T_1] = (10x0.01+ 10x0.5)/20=0.255$

$E[(T_1)^2] = (10x0.0001+10x0.25)/20=0.12505$

$var[T_1] = 0.12505-0.065=0.06005$

$CV^2 = J_1 = 0.06005/0.065 = 0.923$

Next, we compute J_2:

$E[T_1+T_2] = (5x0.02+ 5x1)/10 = 0.51$

$E[(T_1+T_2)^2] = (5x0.0004+5x1)/10 = 0.5002$

$var[T_1+T_2] = 0.5002-0.2601= 0.2401$

$J_2 = 0.2401/2x0.065 = 1.846$

This example clearly illustrates that CV alone is not sufficient to characterize the packet arrival process. Although CV is equal to 0.96, the packet arrival process shown in Figure 3-2 is very bursty.

The index of dispersion for counts (IDC) can be used to observe the variation of the number of packet arrivals in an interval time t and is dimensionless. It is defined as the ratio of the variance of the number of arrivals in time interval t and the mean number of arrivals in time interval t. (Note from the Little's theorem, $E[N_t] = \lambda t$, where λ is the mean packet arrival rate.)

$$\text{Index of Dispersion for Counts (IDC): } I_t = \frac{\text{var}[N_t]}{E[N_t]} \qquad (3\text{-}6)$$

To measure I_t, the time axis is divided into equally spaced intervals with a value t. Denote each interval as a bin. The number of packet arrivals collected in one bin is one sample of N_t. The sample mean and sample variance can be calculated by collecting a sufficient number of samples. $I_t = 0$ for a deterministic process and $I_t = 1$ for the Poisson process. In general, the IDC is not a constant for a renewal process since the number of arrivals in disjoint bins is correlated.

An example of computing IDC is illustrated in Figure 3-3. The number shown in each box is the number of packets contained in each bin. The values of the bin are 0.1 sec and 0.2 sec, respectively.

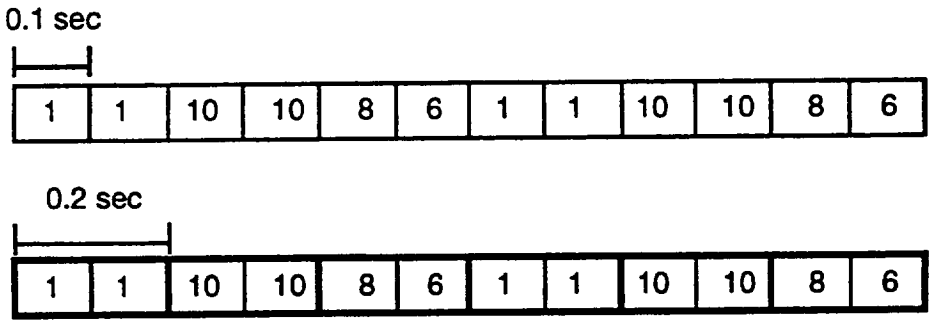

Figure 3-3. Example used to Compute IDC

First, we compute $I_{0.1}$:

$E[N_{0.1}] = 72/12 = 6$

$E[(N_{0.1})^2] = 604/12 = 50.33$

$\text{var}[N_{0.1}] = 50.33\text{-}36 = 14.33$

$I_{0.1} = 2.388$

Next, we compute $I_{0.2}$:

$E[N_{0.2}] = 72/6 = 12$

$E[(N_{0.2})^2] = 1200/6 = 200$

$var[N_{0.2}] = 200\text{-}144 = 56$

$I_{0.2} = 4.66$

By observing that the number of packet counts in a time interval t is less than N only if the sum of N interarrival times is larger than t, it can be proved that the limit of I_t is equal to the limit of J_n [3-3], i.e.,

$$\lim t\to\infty \; It = \lim n\to\infty \; Jn \qquad (3\text{-}7)$$

Remember the IDC is used to compute the variance of packet counts in disjoint intervals. A high IDC suggests that clustering of large packet counts and small packet counts occurs at different bins. The IDC will become constant when there is no correlation among the number of packets in different bins.

The general rule is that the higher the variability and/or burstiness, the higher the queueing delay at the switch.

The index of dispersion can also be used to estimate the parameters for the arrival process. Details will be provided in Section 3.5. It has been argued that to approximate or estimate the parameters of a traffic source, matching the index of dispersion provides a better approximation than matching the (first two) moments of the interarrival times [3-3][3-7][3-13].

3.2 Data Source Models

There are two basic approaches to model a data source. The first is to use a pdf to generate the packet interarrival times. One typical example of this approach is the Poisson traffic source. The second is to dynamically select a pdf based on the state of a Markov chain. The first approach is simple, but it can not capture data burstiness (in time domain). The second approach is more appropriate. The order of presentation of the following queueing models is from simple to complex.

3.2.1 Periodic (Constant Bit Rate) Process

In general, the periodic process is used to model the circuit emulation services. For packetized voice, typically a packet is formed by collecting speech samples in every fixed interval. The voice packet arrivals becomes periodic. The period is equivalent to the packetization delay. For file transfer, end-to-end flow control is necessary to regulate the transfer rate. File transfer can also be modeled as periodic process. The peak rate of the periodic process is the same as the average rate. There is no statistical gain of multiplexing multiple periodic processes. The periodic process distribution is characterized by the pdf of the packet interarrival time:

$$a(t) = \begin{cases} 1 & t=1/\lambda \\ 0 & \text{otherwise} \end{cases} \qquad (3\text{-}8)$$

where the arrival rate λ is a constant.

The value of CV is 0, and $J_n = 0$ for all n, and $I_t = 0$ for all t.

3.2.2 Poisson Process

The Poisson process is a renewal process. There are two ways of describing a Poisson process. The first is to use the probability for the number of arrivals within a given interval. The other is to use the interarrival time.

The probability that there are N arrival packets within time t is

$$P_N(t) = \frac{(\lambda t)^N e^{-\lambda t}}{N!}, N \geq 0, t \geq 0. \tag{3-9}$$

The mean and variance can be computed by using the probability generating function G(z), where z is a complex variable and $|z| \leq 1$.

$$G(z) = E[z^N] = \sum_{N=0} z^N \frac{(\lambda t)^N e^{-\lambda t}}{N!}$$

$$= e^{-\lambda t} \sum_{N=0} \frac{(z\lambda t)^N}{N!} = e^{-\lambda t}\ e^{\lambda z t}. \tag{3-10}$$

From the property of the probability generating function,

$$G(z)|_{z=1} = 1, \frac{\partial}{\partial z} G(z)|_{z=1} = E[N_t], \text{ and } \frac{\partial^2}{\partial z^2} G(z)|_{z=1} = E[N_t^2] - E[N_t].$$

Therefore,

$$E[N_t] = \lambda t \text{ and } (\sigma_{Nt})^2 = \lambda t. \tag{3-11}$$

The Poisson arrival process has exponential interarrival times. It should be mentioned that the exponential distribution is the only continuous distribution which has the memoryless property [3-22]. That is to say the future of an exponentially distributed random variable is independent of the past history of the variable. The only discrete distribution which has the same property is the Geometric distribution. The Geometric distribution will be described later.

The probability cumulative function of interarrival time is

$$A(t) = 1 - P_0(t) = 1 - e^{-\lambda t} \tag{3-12}$$

where $t \geq 0$. The probability density function is

$$a(t) = \lambda e^{-\lambda t} \tag{3-13}$$

where $t > 0$. The mean and variance of the exponential distribution can be computed using the Laplace transform of the distribution.

$$A(s) = E[e^{-st}] = \int_0^\infty e^{-st}\, a(t)\, dt, \text{ where s is a complex variable and } A(s)_{|s=0} = 1.$$

$$A(s) = \int_0^\infty e^{-st}\, \lambda e^{-\lambda t}\, dt = \lambda \int_0^\infty e^{-(s+\lambda)t}\, dt = \frac{\lambda}{\lambda+s}. \tag{3-14}$$

A key use of this transform is its moment generating property, i.e.,

$$\frac{\partial^k}{\partial s^k} A(s)_{|s=0} = (-1)k\, E[t^k]. \text{ Since } \frac{\partial}{\partial s} A(s) = \frac{-\lambda}{(\lambda+s)^2} \text{ and } \frac{\partial^2}{\partial s^2} A(s) = \frac{2\lambda(\lambda+s)}{(\lambda+s)^4},$$

$$E[t] = \frac{1}{\lambda}, \text{ var}[t] = (\sigma_t)^2 = \frac{1}{\lambda^2}, \text{ and } E[t^2] = \frac{2}{\lambda^2} \tag{3-15}$$

For the Poisson process, the packet arrivals are independent. The three traffic characterization measures are: CV = 1, $J_n = 1$ for all n and $I_t = 1$ for all t.

In the past, the Poisson process is widely accepted for data traffic model. The major reason is that superposition of a larger number of two-state Markov processes will converge to the Poisson process [3-11]. The two-state Markov process will be described later. However, the accuracy of using the Poisson process to model the real-world data traffic has been questioned by many researchers. Clearly, the Poisson process is not capable of modeling a traffic pattern with a high variation or a traffic pattern with correlation between successive packets. Experiments were performed in Reference 3-15 to measure and analyze the traffic pattern on an Ethernet LAN. The results show that the packet arrival process is not Poisson.

Since the destination directed packet switching (DDPS) satellite network employs slotted operation, the continuous Poisson process must be converted to a discrete Poisson process. Since the interarrival time of the continuous Poisson process follows the exponential distribution, simultaneous arrivals of packets are possible. To generate the discrete Poisson process, the continuous Poisson process is fed into a FIFO. The FIFO generates packets at discrete slot time. The function of the FIFO is two-fold. One is to convert continuous packet arrival times into discrete packet arrival times, where the discrete unit is the packet slot time. The other is to convert simultaneous arrivals into a batch of arrivals. For example, simultaneous arrivals of 3 packets becomes a batch of 3 packets with no spacing between packets.

A simulation model is developed to analyze the characteristics of the Poisson process. The parameters used in the simulation is listed in Table 3-1. To completely specify the Poisson process, only the mean packet interarrival time is required. In this example, a mean link utilization of 0.9 is translated into a mean packet interarrival time of 0.00111 sec.

Table 3-1. Simulation Parameters for Poisson Process

PARAMETER	VALUE
link speed	2.048 Mbit/sec
packet slot time	0.001 sec
packet interarrival time	0.00111 sec
mean link utilization	0.9

The simulated CV is 0.45. The simulated CV is less than the theoretical value 1. The reason is that the continuous Poisson process has been converted into a discrete version. Figure 3-4 shows the simulated interarrival time pdf, log scale interarrival time pdf, the IDC and the IDI curves.

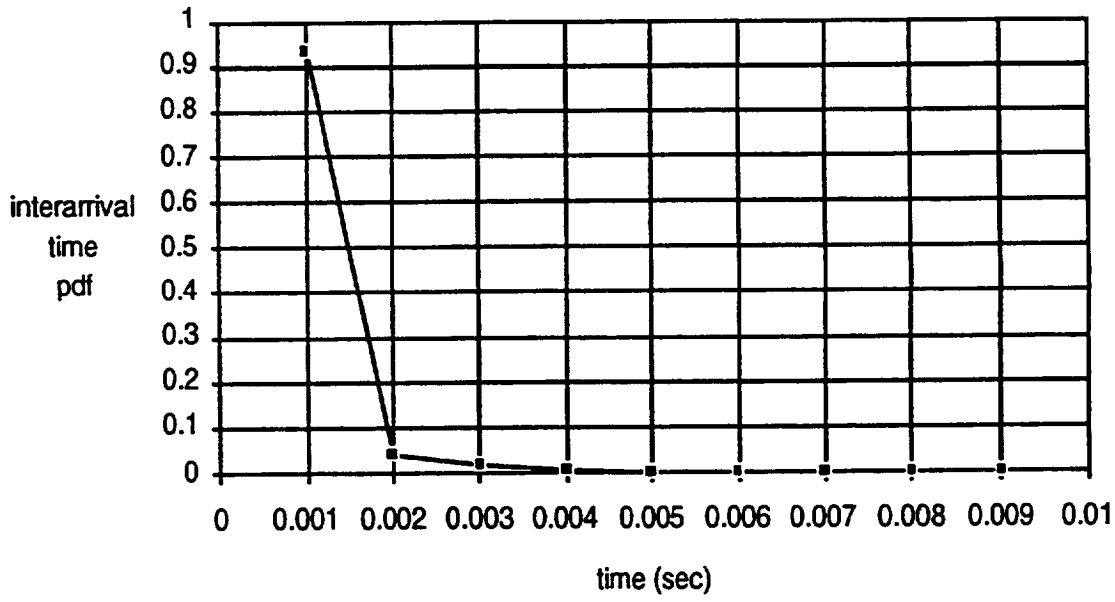

Figure 3-4 (a). Packet Interarrival Time pdf for Poisson Process

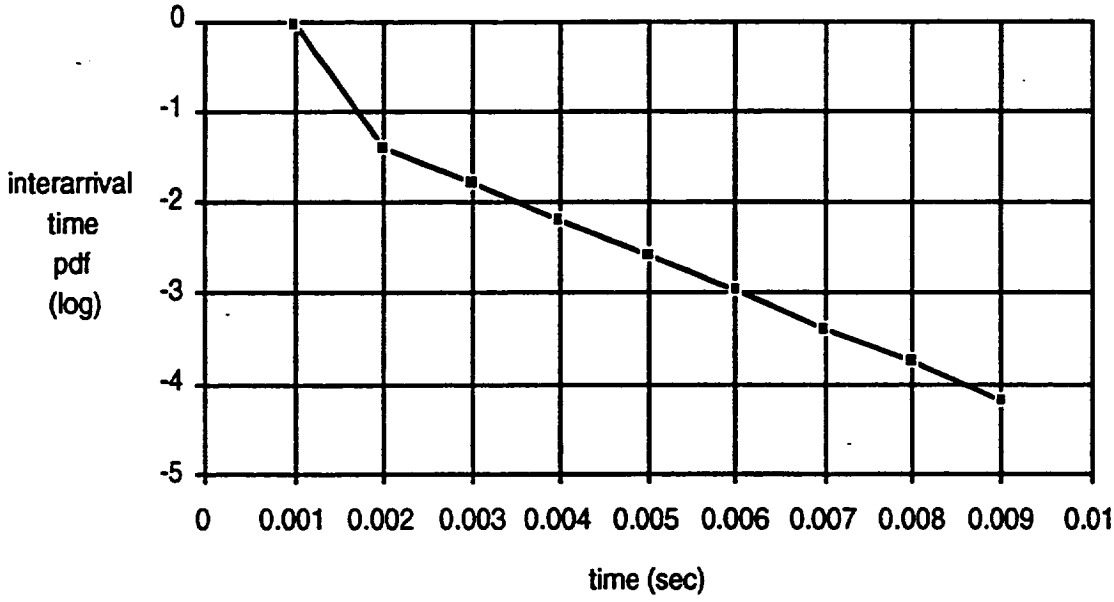

Figure 3-4 (b). Log Scale Packet Interarrival Time pdf for Poisson Process

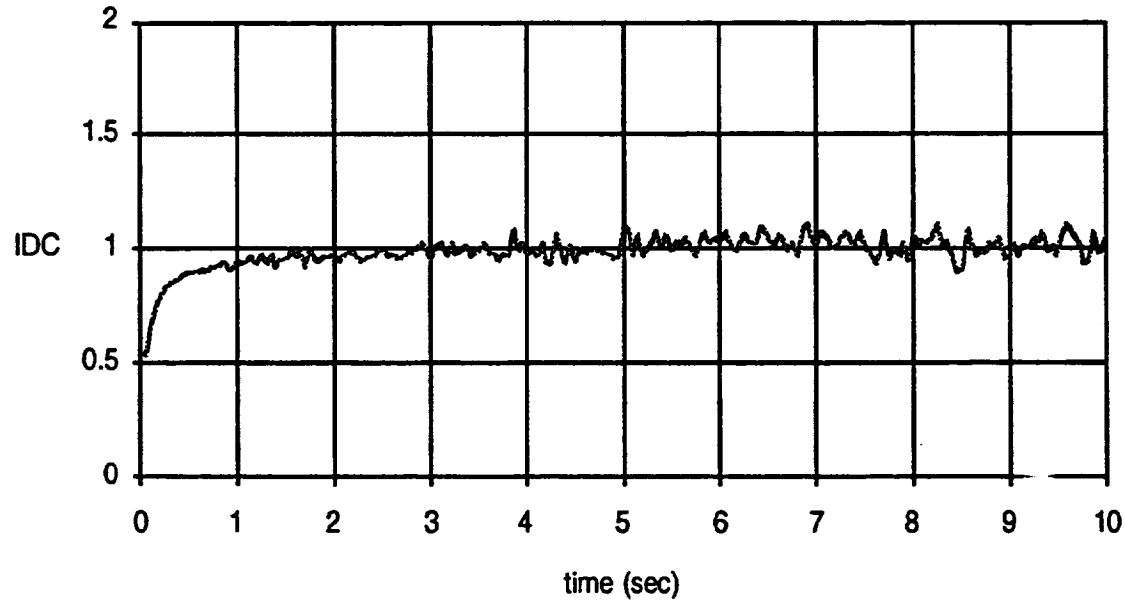

Figure 3-4 (c). Index of Dispersion for Counts (IDC) for Poisson Process

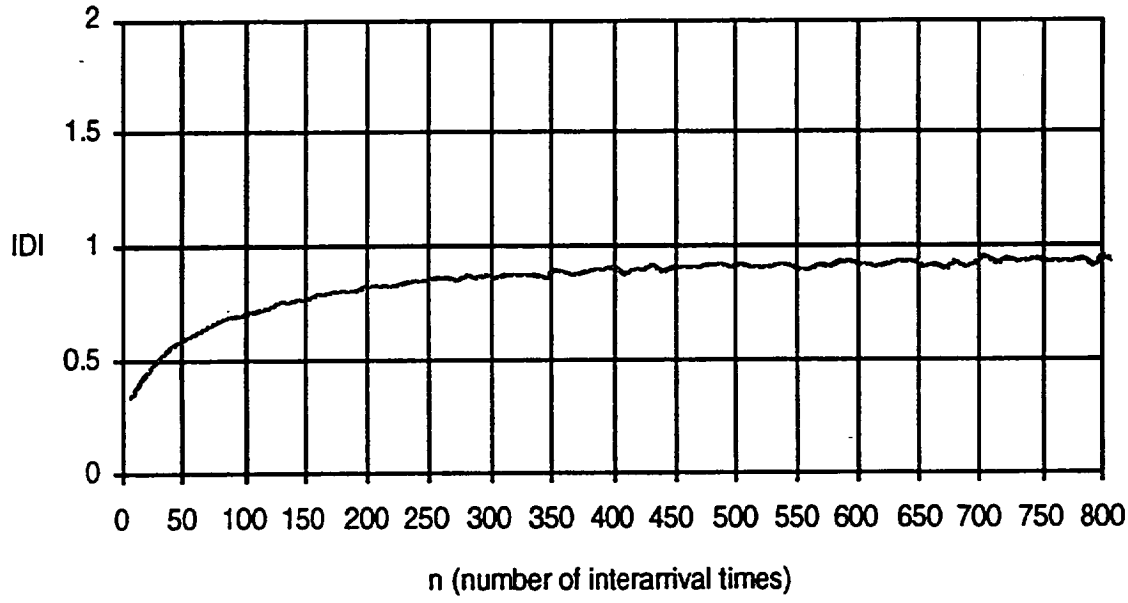

Figure 3-4 (d). Index of Dispersion for Intervals (IDI) for Poisson Process

3.2.3 Geometric Distribution

The Poisson process is used to model the continuous data traffic source, and geometric distribution is used to model the discrete data traffic source. Define the random variable X to be the number of Bernoulli trials to achieve the first success.

The Bernoulli process is characterized by independent trials, and each trial has only two outcomes (success or fail). Let the probability of success be p. The Bernoulli distribution is defined as

$$P(x) = \begin{cases} p & \text{for } x=1 \text{ (success)} \\ 1-p & \text{for } x=0 \text{ (fail)} \end{cases} \quad (3\text{-}16)$$

The mean and the variance of the Bernoulli process are expressed as

$$E[x] = 1\,p + 0\,(1-p) = p \quad (3\text{-}17)$$

$$\text{var}[x] = [1^2\,p + 0^2\,(1-p)] - p^2 = p\,(1-p). \quad (3\text{-}18)$$

For the Geometric process, the event of success is defined as the arrival of a packet, and the number of trials is equivalent to the packet interarrival time (in slots). The pdf of X, $f_X(n)$, is given as

$$f_X(n) = p\,(1-p)^{n-1},\ n \geq 1. \quad (3\text{-}19)$$

The mean of random variable X (the interarrival time) can be derived either using the probability generating function G(z) or the following procedure.

$$E[X] = \sum_{n=1}^{\infty} n\, p(1-p)^{n-1} = \frac{p}{(1-p)} \sum_{n=0}^{\infty} n\,(1-p)^n = \frac{p}{(1-p)}(1-p)\frac{\partial}{\partial(1-p)} \sum_{n=0}^{\infty}(1-p)^n$$

$$E[X] = p\,\frac{\partial}{\partial(1-p)}\frac{1}{1-(1-p)} = \frac{1}{p} \qquad (3\text{-}20)$$

To obtain var[X], the $E[X^2]$ needs to be derived first.

$$E[X^2] = \sum_{n=1}^{\infty} n^2\, p(1-p)^{n-1} = \frac{p}{(1-p)} \sum_{n=0}^{\infty} n^2\,(1-p)^n$$

$$= \frac{p}{(1-p)}\left[(1-p)^2\,\frac{\partial^2}{\partial(1-p)^2} \sum_{n=0}^{\infty}(1-p)^n + \sum_{n=0}^{\infty} n\,(1-p)^n\right]$$

$$= \frac{p}{(1-p)}\left[(1-p)^2\,\frac{1}{p^3} + \frac{1-p}{p^2}\right] = \frac{2-p}{P2}$$

Therefore,

$$\text{var}[X] = E[X^2] - E[X]^2 = \frac{(1-p)}{p^2} \quad \text{and} \quad CV = \sqrt{1-p} \qquad (3\text{-}21)$$

The Geometric distribution has been successfully used to describe the randomness of traffic sources. It has been suggested that the arrival process for a single source in a discrete-time (slotted) system with low-speed services (such as 32-kbit/sec voice and data) can be considered to be pure random [3-23]. The random traffic has also been suggested to model the background noise of a DDPS.

A simulation model is developed to analyze the characteristics of the Geometric process. The parameters used in the simulation are listed in Table 3-2. To completely specify the Geometric process, only the probability of success p (the link mean utilization) is required. A mean link utilization of 0.9 is translated into a mean packet interarrival time of 0.00111 sec. The simulated CV is 0.316. The simulated interarrival time pdf, log scale interarrival time pdf, the IDC and the IDI curves are shown in Figure 3-5.

Table 3-2. Simulation Parameters for Geometric Process

PARAMETER	VALUE
link speed	2.048 Mbit/sec
packet slot time	0.001 sec
packet interarrival time	0.00111 sec
mean link utilization	0.9

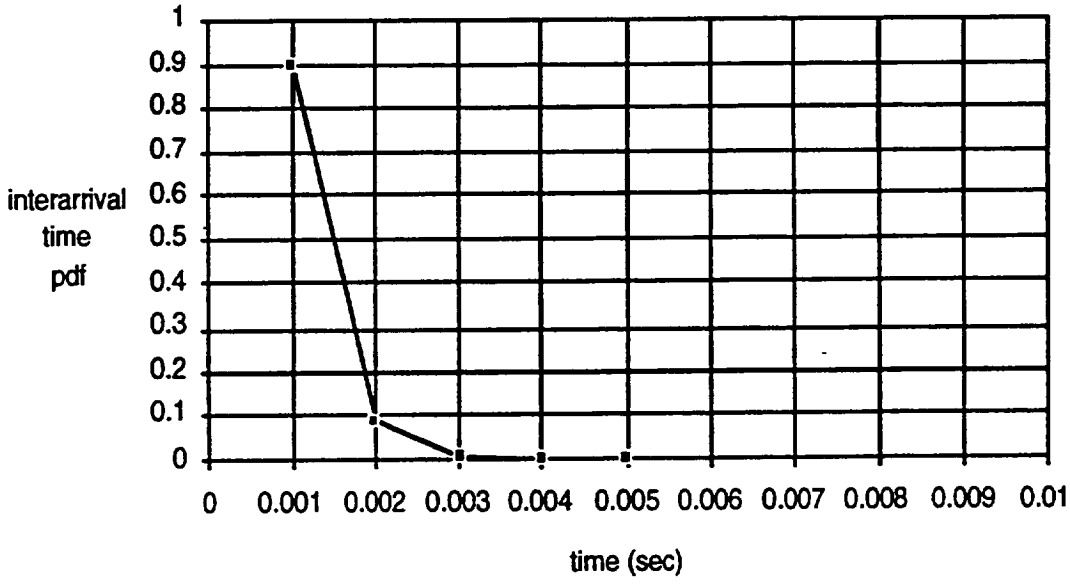

Figure 3-5 (a). Packet Interarrival Time pdf for Geometric Process

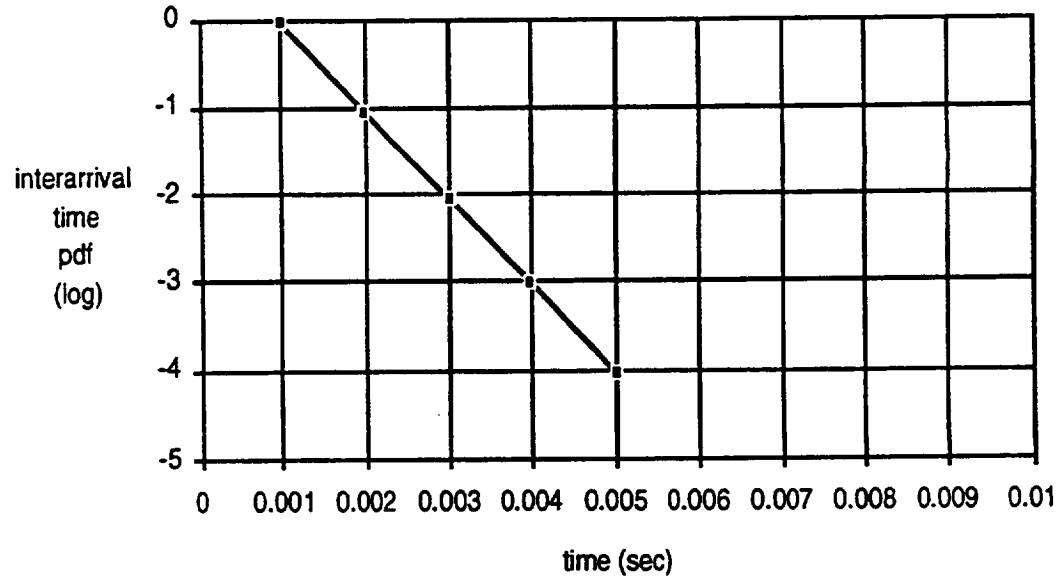

Figure 3-5 (b). Log Scale Packet Interarrival Time pdf for Geometric Process

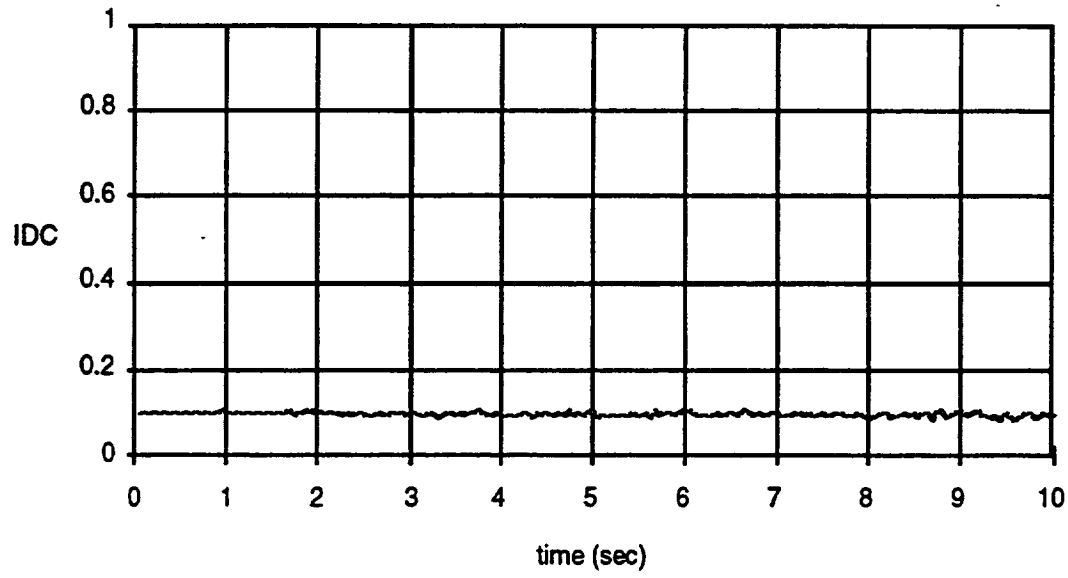

Figure 3-5 (c). Index of Dispersion for Counts (IDC) for Geometric Process

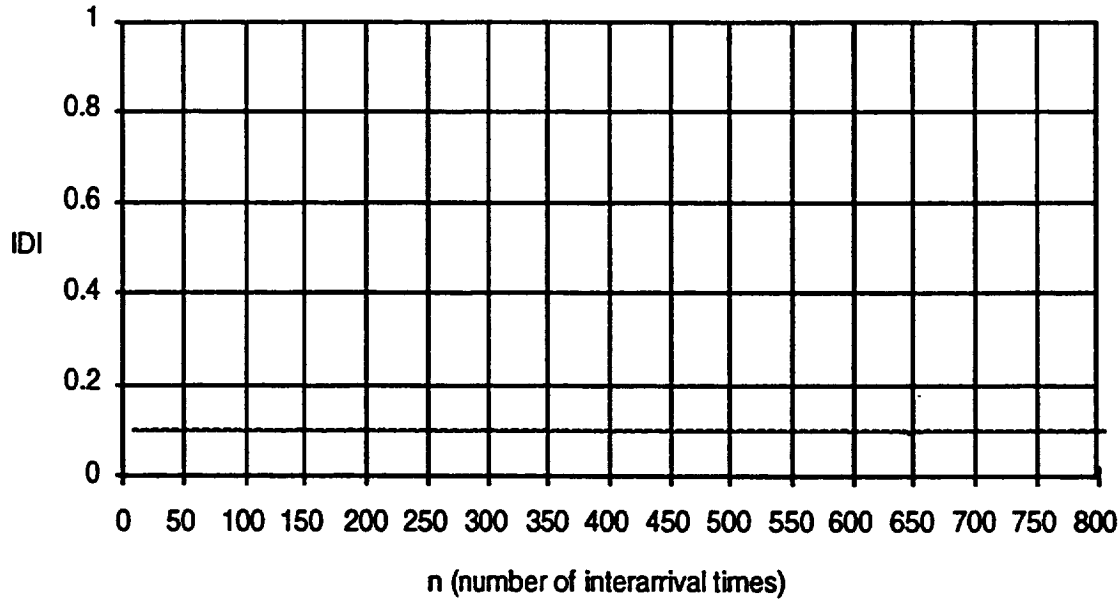

Figure 3-5 (d). Index of Dispersion for Intervals (IDI) for Geometric Process

3.2.4 Weibull Distribution

Weibull distribution was suggested in Reference [3-15] to model the bursty traffic. The general form of Weibull distribution is

$$\frac{\beta\, t^{\beta-1}}{\alpha^{\beta-1}} e^{-(t/\alpha)\beta} \qquad (3\text{–}22)$$

Weibull distribution can be used to fit different packet interarrival time distributions by changing α and β. For example, when $\beta = 1$, it becomes an exponential distribution and when $\beta = 2$, it becomes a Rayleigh distribution.

3.2.5 Batch Poisson Process

The batch Poisson process is an extension of the Poisson process. Instead of generating single packet arrivals, the arrivals of packets are in the form of batches. The size of the batch is a random number (r) and the interarrival time of batches follows an exponential distribution. The number of packet arrivals in an interval t is

$$\sum_{i=1}^{N(t)} r_i.$$

The average number of packet arrivals in an interval t is $E[N(t)]E[r] = \lambda t\, E[r]$. The batch Poisson process may be too bursty for data traffic since the packet streams are most likely to be spaced out in time [3-7].

The IDC of a batch Poisson process is expressed as

$$I_t = \frac{\mathrm{var}(r_i)}{E[r_i]} + E[r_i].$$

Assume the batch size follows the Geometric distribution. Then

$$f_x(n) = (1-p)\, p^{n-1},\ n \geq 1.$$

The average batch size is $\frac{1}{1-p}$. The IDC becomes $\frac{1+p}{1-p}$. If the average batch size is b, then $p = \frac{b-1}{b}$.

A simulation model is developed to analyze the characteristics of the batch Poisson process. The parameters used in the simulation are listed in Table 3-3. To completely specify the batch process, the mean packet interarrival time and the average batch size need to be specified.

Table 3-3. Simulation Parameters for Batch Poisson Process

PARAMETER	VALUE
link speed	2.048 Mbit/sec
packet slot time	0.001 sec
packet interarrival time	0.006666 sec
batch size	6
mean link utilization	0.9

The theoretical value of IDC is 11. The simulated CV is 1.095. The simulated interarrival time pdf, log scale interarrival time pdf, the IDC and the IDI curves are shown in Figure 3-6.

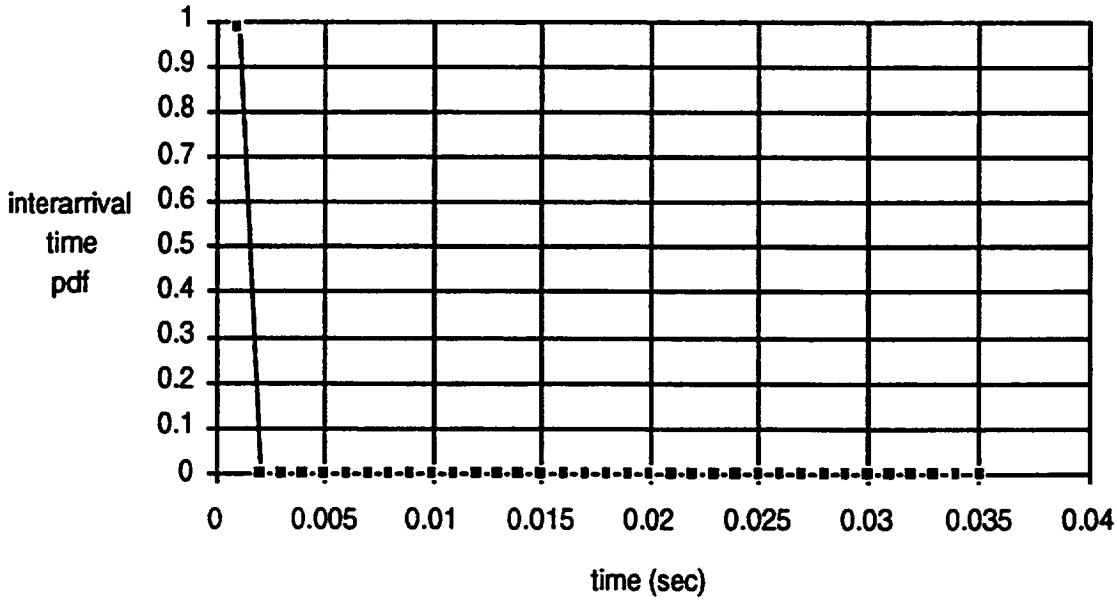

Figure 3-6 (a). Packet Interarrival Time pdf for Batch Poisson Process

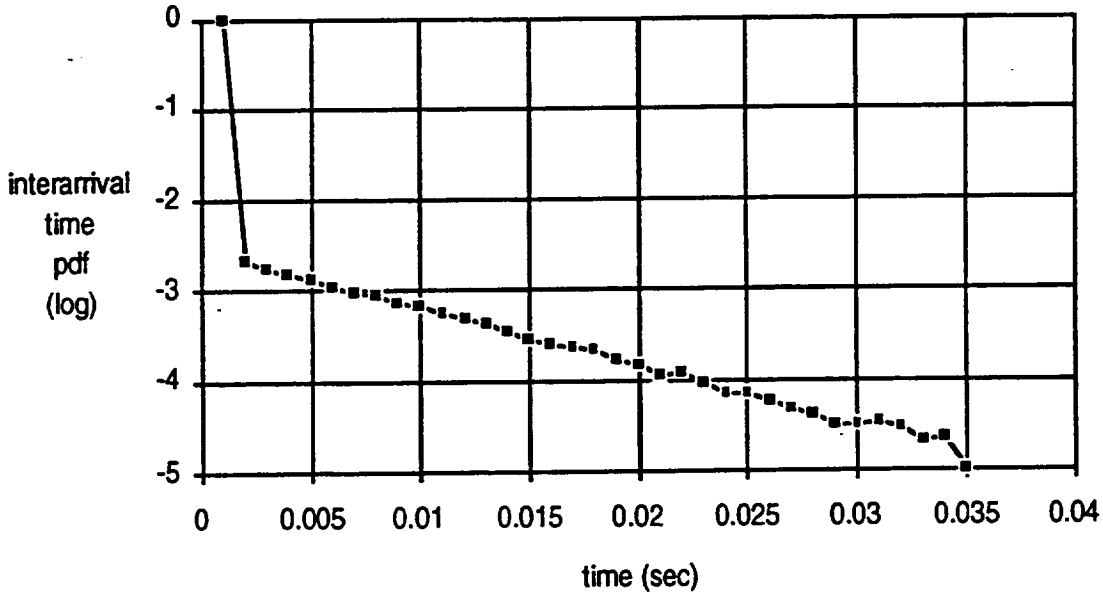

Figure 3-6 (b). Log Scale Packet Interarrival Time pdf for Batch Poisson Process

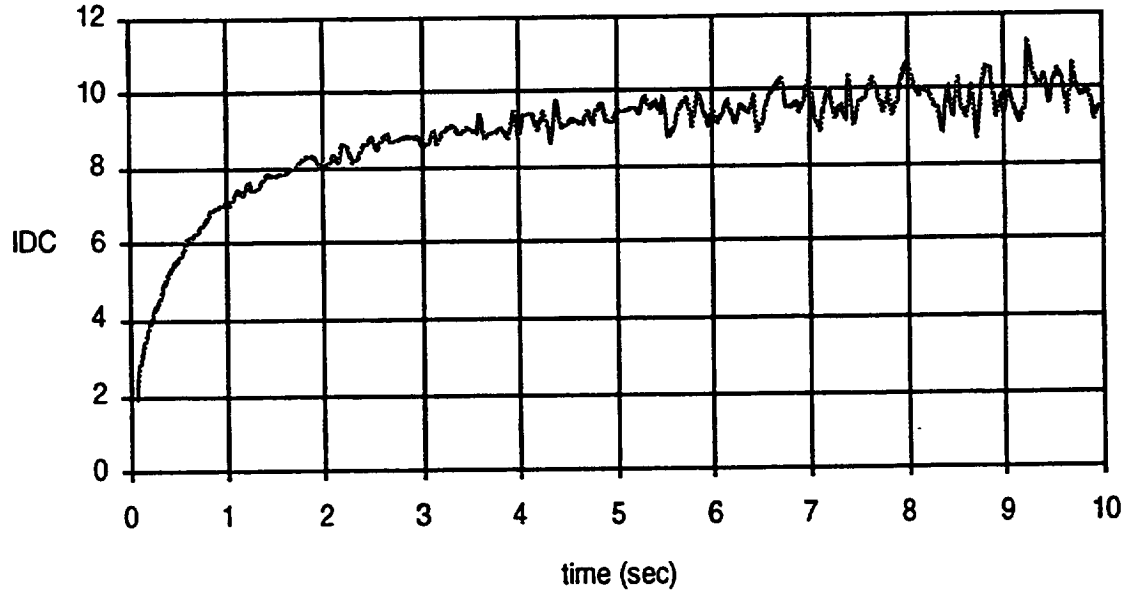

Figure 3-6 (c). Index of Dispersion for Counts (IDC) for Batch Poisson Process

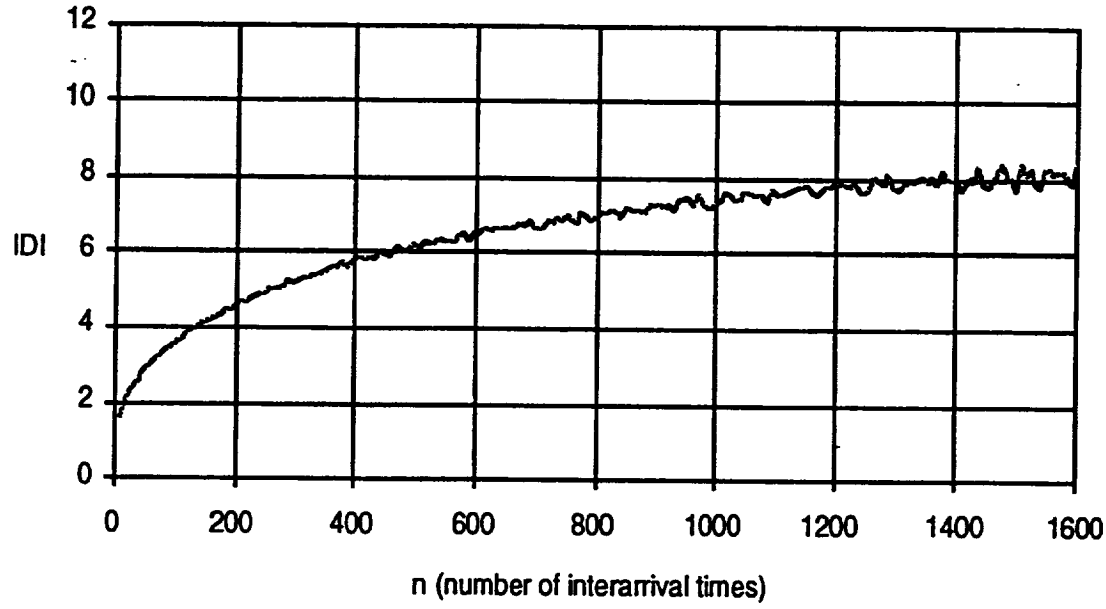

Figure 3-6 (d). Index of Dispersion for Intervals (IDI) for Batch Poisson Process

3.2.6 Hyperexponential Interarrival Time Process

The hyperexponential interarrival time process is a mixture of independent Poisson processes. The hyperexponential interarrival time process is a renewal process. A hyperexponential distribution of order k, H_k, is the weighted sum of k exponential distributions, i.e.,

$$P(t \le t_0) = \sum_{i=1}^{k} \alpha_i \, (1 - e^{-\lambda_i t_0}) \qquad (3\text{-}23)$$

where $\alpha_i > 0$ and $\sum_{i=1}^{k} \alpha_i = 1$.

Consider H2, where $\alpha_2 = 1 - \alpha_1$.

$$P(t \le t_0) = \alpha_1 \, (1 - e^{-\lambda_1 t_0}) + (1 - \alpha_1) \, (1 - e^{-\lambda_2 t_0}) \qquad (3\text{-}24)$$

The mean and the variance of interarrival time for H2 were derived in Reference 3-3 and given as follows:

$$E\,[H2] = \frac{\alpha_1 \lambda_2 + (1 - \alpha_1) \lambda_1}{\lambda_1 \lambda_2} \qquad (3\text{-}25)$$

$$\text{var}[H2] = \frac{2(1-\alpha_1)(\lambda_1)^2 + 2\alpha_1(\lambda_2)^2 - ((1-\alpha_1)\lambda_1 + \alpha_1\lambda_2)^2}{(\lambda_1)^2(\lambda_2)^2} \qquad (3\text{-}26)$$

H2 was used in [3-9] to model the interarrival time of a bursty multiplexer traffic source. Although the Hyperexponential interarrival time process can model a traffic pattern with a high variation, it can not model a traffic pattern with correlation between successive packets.

Note that the hyperexponential interarrival time distribution is the sum of two independent exponential distributions. This should not be confused with obtaining the distribution for a random variable z, where z = x + y and x and y are independent and exponentially distributed. The pdf for z is the convolution of the pdfs of x and y.

There is another subtle difference between the exponential distribution and the hyperexponential distribution. For the exponential distribution, prob[$t \le x+t_0 \mid t>t_0$] = prob[$t \le x$] = $1 - e^{-\lambda x}$. However, for the hyperexponential distribution, prob[$t \le x+t_0 \mid t>t_0$] is monotonically increasing with t_0 [3-15]. This means the longer the time elapses from the last arrival, the expected time for the next arrival is also longer.

A simulation model is developed to analyze the characteristics of the hyperexponential interarrival time process. The parameters used in the simulation are listed in Table 3-4. To completely specify the H2 process, two mean packet interarrival times and the coefficient need to be specified.

Table 3-4. Simulation Parameters for Batch Poisson Process

PARAMETER	VALUE
link speed	2.048 Mbit/sec
packet slot time	0.001 sec
packet interarrival time 1	0.001076 sec
packet interarrival time 2	0.00125 sec
coefficient	0.8
mean link utilization	0.9

The simulated CV is 1.196. The simulated interarrival time pdf, log scale interarrival time pdf, the IDC and the IDI curves are shown in Figure 3-7.

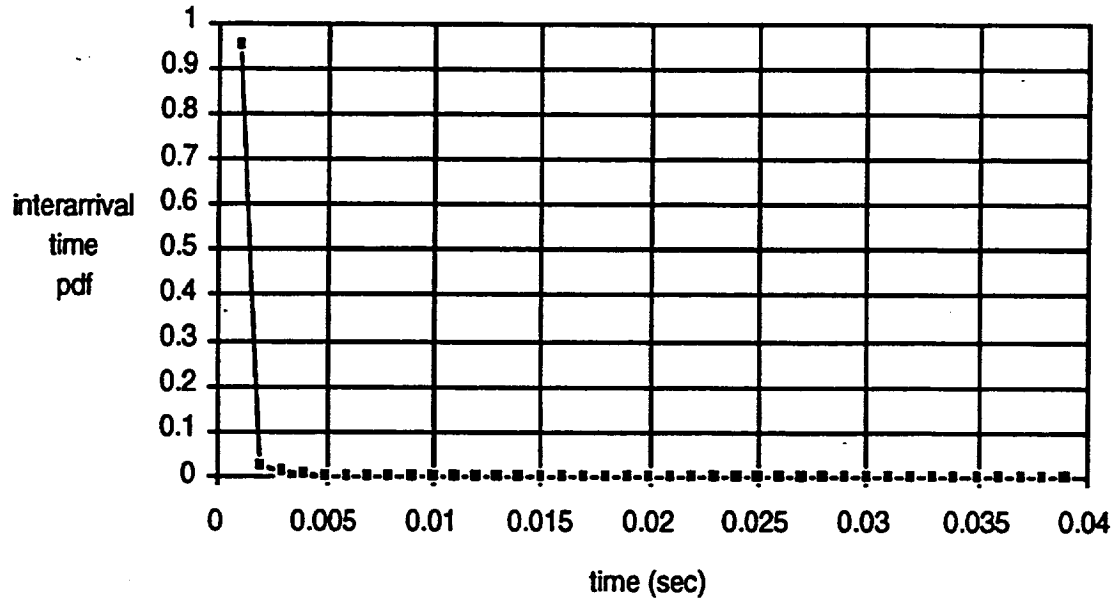

Figure 3-7 (a). Packet Interarrival Time pdf for Hyperexponential Interarrival Time Process

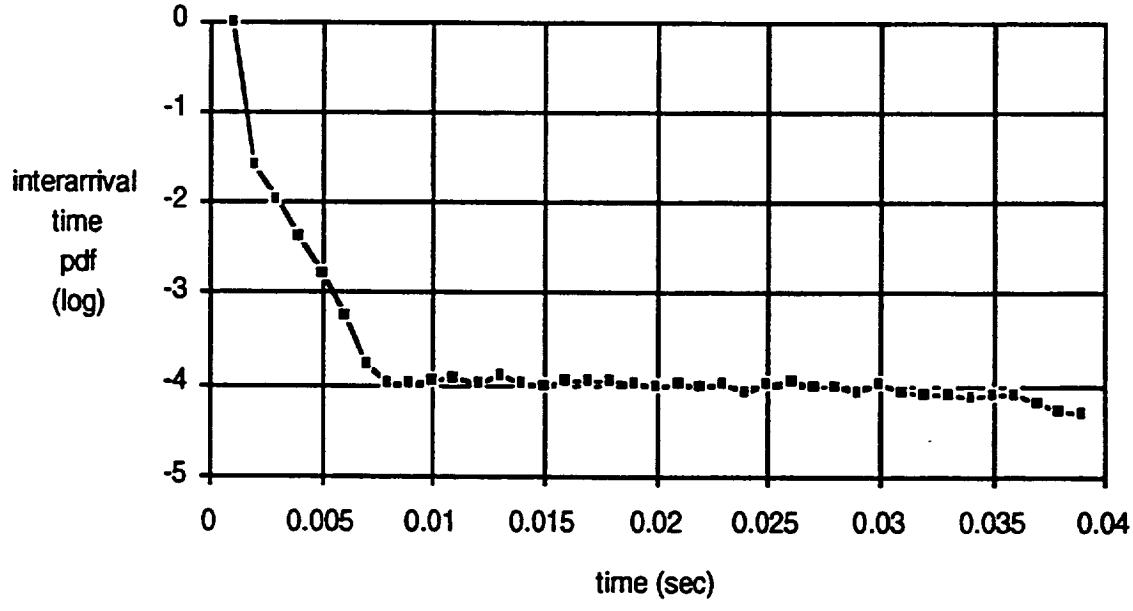

Figure 3-7 (b). Log Scale Packet Interarrival Time pdf for Hyperexponential Interarrival Time Process

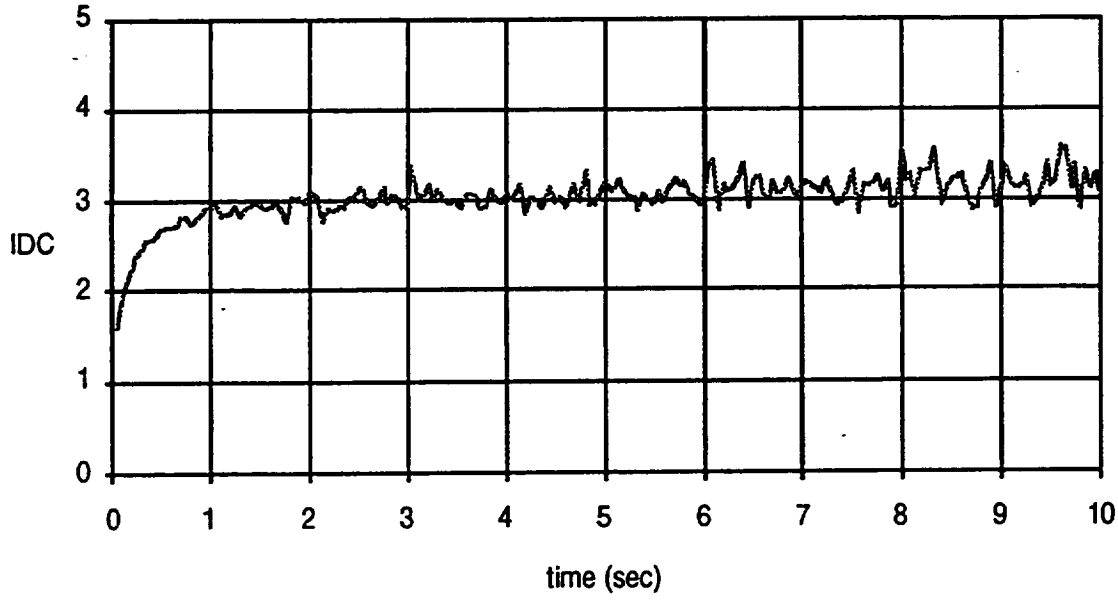

Figure 3-7 (c). Index of Dispersion for Counts (IDC) for Hyperexponential Interarrival Time Process

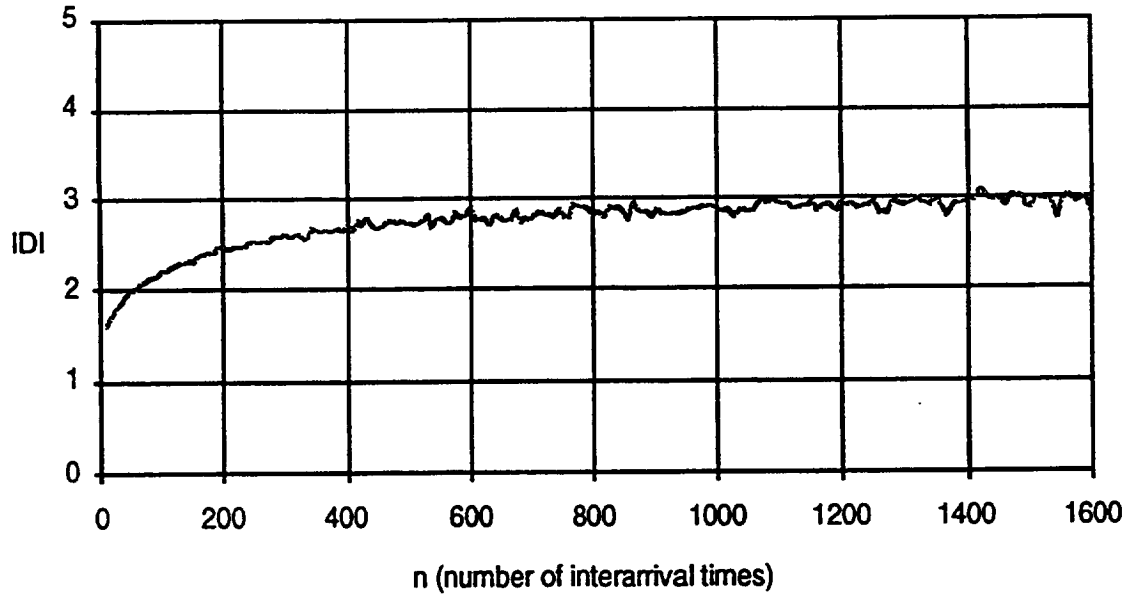

Figure 3-7 (d). Index of Dispersion for Intervals (IDI) for Hyperexponential Interarrival Time Process

3.2.7 Markov-Modulated Poisson Process

The Markov-Modulated Poisson process (MMPP) was originally used to model the errors in the (mobile) fading communications channels [3-20]. The fading channels are modeled to be either in a normal state (with low BER) or an error state (high BER). The normal state and error state of a fading channel alternates across time.

The error bursts of the fading channel are analogous to the traffic bursts. Therefore, the MMPP has been suggested to be used to model the aggregate traffic from multiple sources. In an n-state MMPP, the transition between states is based on the Markov model. When the Markov model is in state i, the arrival process is Poisson with arrival rate λ_i. The n-state MMPP is completely defined by the transition matrix and the arrival rates (see Figure 3-8).

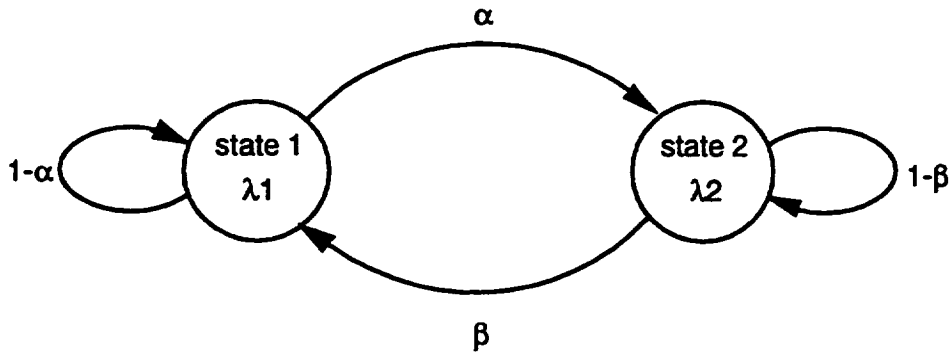

Figure 3-8. Two-State MMPP

There are two options of choosing the Markov model: the discrete time or the continuous time. For a discrete time Markov model, the transitions between states only occur at discrete time. If the traffic generator generates fixed size packets, then the discrete time Markov model should be used; if it generates variable size packets, then the continuous time model should be used.

The next issue is to choose the number of states. At different states, the arrival rate λ of the Poisson process is different. Most researchers have chosen the two-state Markov models, since this model is easy to use for analysis and takes into consideration the correlation of interarrival times between successive packets. The capability of the two-state MMPP of modeling the burstiness and correlation between successive packets for data traffic will be illustrated using a simulation model later.

The two-state MMPP is the most generalized and widely used traffic model to represent the bursty traffic and a superposition of packetized voice and data traffic. The two-state MMPP is capable of modeling burstiness and correlation of the packet arrival process and is simple for analysis. This model is recommended for traffic simulator implementation.

At each slot time, the source model is either in state 1 or in state 2. In state 1, the traffic source will generate information packets with rate λ_1, and in state 2, the traffic source will generate information packets with rate λ_2. With the assumption of the two-state Markov chain model, the periods of states 1 and 2 will be geometrically distributed, with respective mean values $E[T_1]$ and $E[T_2]$ slot times.

Let P_1 (or P_2) be the probability that the source is in state 1 (or state 2). P_1 (or P_2) can be expressed either using $E[T_1]$ and $E[T_2]$ or α and β. It is clear that

$$P_1 = \frac{E[T_1]}{E[T_1] + E[T_2]} \quad \text{and} \quad P_2 = \frac{E[T_2]}{E[T_1] + E[T_2]} \tag{3-27}$$

By solving the stationary equation

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$$

, the equilibrium state probabilities can be obtained as

$$P_1 = \frac{\beta}{\alpha+\beta} \text{ and } P_2 = \frac{\alpha}{\alpha+\beta}. \tag{3-28}$$

Burstiness can be characterized by 1) the ratio of the peak rate to the average rate, 2) how frequently the burst occurs, i.e., P_1 (assuming $\lambda_1 > \lambda_2$), 3) how long the burst last, i.e., T_1, 4) CV, 5) IDI and 6) IDC.

The mean arrival rate, the mean squared arrival rate, the variance of the arrival rate, and the average number of packets in an interval t for a two-state MMPP are given in [3-13]

$$E[\lambda] = P_1 \lambda_1 + P_2 \lambda_2 = \frac{\lambda_1 E[T_1] + \lambda_2 E[T_2]}{E[T_1]+E[T_2]} \tag{3-29}$$

$$E[(\lambda)^2] = P_1 (\lambda_1)^2 + P_2 (\lambda_2)^2 = \frac{(\lambda_1)^2 E[T_1] + (\lambda_2)^2 E[T_2]}{E[T_1]+E[T_2]} \tag{3-30}$$

$$var[\lambda] = P_1 P_2 (\lambda_1 - \lambda_2)^2 \tag{3-31}$$

$$E[N_t] = E[\lambda] t. \tag{3-32}$$

The I_t has been derived in Reference [3-13], the limit of I_t is given as

$$I_\infty = 1 + \frac{2(E[T_1]E[T_2])^2 (\lambda_1-\lambda_2)^2}{(E[T_1]+E[T_2])^2 (E[T_1]\lambda_1+E[T_2]\lambda_2)} \tag{3-33}$$

Let $r = \frac{1}{E[T_1]} + \frac{1}{E[T_2]}$. Then

$$I_t = I_\infty - \frac{2(E[T_1]E[T_2])^3(\lambda_1-\lambda_2)^2}{(E[T_1]+E[T_2])^3(E[T_1]\lambda_1+E[T_2]\lambda_2)t}(1-e^{-rt}) \quad (3\text{-}34)$$

and

$$\frac{I_\infty - I_t}{I_\infty - 1} = \frac{1-e^{-rt}}{rt} \quad (3\text{-}35)$$

In Reference 4-6, a K state MMPP is proposed. The arrival rate at state j is given as

$$\lambda_j = (j+1)\lambda_0, \quad (3\text{-}36)$$

where λ_0 is a constant and j is based on the birth-death model (M/M/1/K) with birth rate γ_j and death rate μ_j. Let

$\gamma_j = \gamma$, for $0 \le j < K$ and $\gamma_K = 0$.

$\mu_j = \mu$, for $0 < j \le K$ and $\mu_0 = 0$.

Let $\rho = \frac{\gamma}{\mu}$. The probability that the process is in state j is given as

$$\frac{1-\rho}{1-\rho^{K+1}}\rho^j.$$

The average arrival rate is given as

$$\left(\frac{\rho}{1-\rho^{K+1}}\,\frac{1+K\rho^{K+1}-(K+1)\rho^K}{1-\rho}+1\right)\lambda_0.$$

In Reference 3-14, the superposition of homogeneous on-off processes was approximated by a two-state MMPP. The characteristics of the on-off process will be described later. The main arguments to support this statement are: 1) the two-state MMPP is a correlated and nonrenewal process and 2) the model is to analyze the performance of an ATM multiplexer. From the viewpoint of a multiplexer, the rate of the aggregate input traffic is either higher or lower than the service rate. Therefore, the two states of the MMPP can be used to represent the overload state and the underload state, respectively.

Note that the superposition of the MMPP with an independent Poisson process with λ_3 also results in an MMPP with $\lambda_1 = \lambda_1 + \lambda_3$ and $\lambda_2 = \lambda_2 + \lambda_3$. The superposition of independent MMPPs is also an MMPP [3-30].

In Reference 3-3, real traffic measurements are performed for SUN workstations connected by an Ethernet. The measurement result was fit into a two-state MMPP. The fitting procedure will be described later. The parameter values obtained for the two-state MMPP are listed in Table 3-5.

Table 3-5. Typical Parameter Values for Two-State MMPP

λ_1 (1/sec)	λ_2 (1/sec)	$E[T_1]$ (sec)	$E[T_2]$ (sec)	link rate (Mbit/sec)	packet size (byte)
9.61	77.37	14.7	196.0	10	1500

In Reference 3-16, the MMPP is used to model ATM traffic sources. The normal state $E[T_1]$ is in the range of 0.1 sec and 0.5 sec and λ = 6407.6 cell/sec (= 2.7168 Mbit/sec divided by 424 bits). The burst state sends out cells in a much higher, constant rate than the normal state. To facilitate simulation, the burst state duration can use any distribution.

A simulation model is developed to analyze the characteristics of the two-state MMPP. The parameters used in the simulation are listed in Table 3-6. To completely specify the two-state MMPP, four parameters are required.

Table 3-6. Simulation Parameters for Two-State MMPP

PARAMETER	VALUE
link speed	2.048 Mbit/sec
packet slot time	0.001 sec
mean bursty state duration ($E[T_1]$)	2.5 sec
mean bursty state interarrival time ($\frac{1}{\lambda_1}$)	0.00102 sec
mean normal state duration ($E[T_2]$)	0.25 sec
mean normal state interarrival time ($\frac{1}{\lambda_2}$)	0.01 sec
mean link utilization	0.9

The simulated CV is 1.245. The burstiness of the two-state MMPP is shown in Figure 3-9a for the first 50 sec. As seen in this figure, the bursty state and the normal state alternates across time. The simulated IDC and computed curves of the two-state MMPP are shown in Figure 3-9 (d). The simulation time is 2000 sec. To obtain a smoother curve for the simulated IDC, a longer simulation time is required. The limit of IDC is computed using Equation 3-21, and the result is I_∞ = 33.323. The simulated IDI curve is shown in Figure 3-9 (e). It can be verified that the limit of IDC is equal to that of IDC by comparing Figure 3-9d and Figure 3-9e.

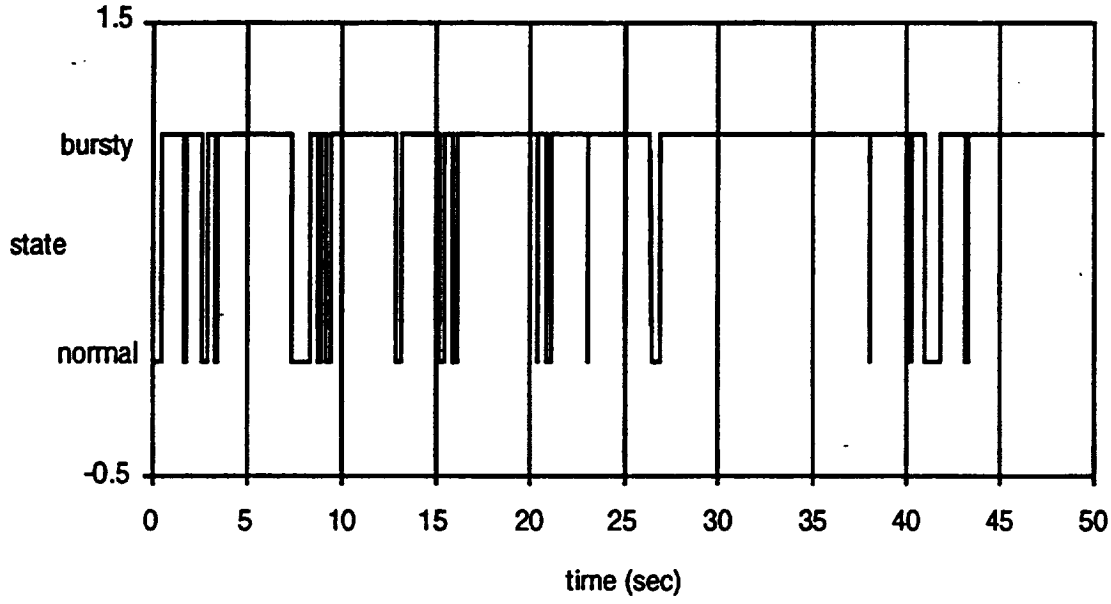

Figure 3-9 (a). Traffic Characteristics of Two-State MMPP in Time Axis

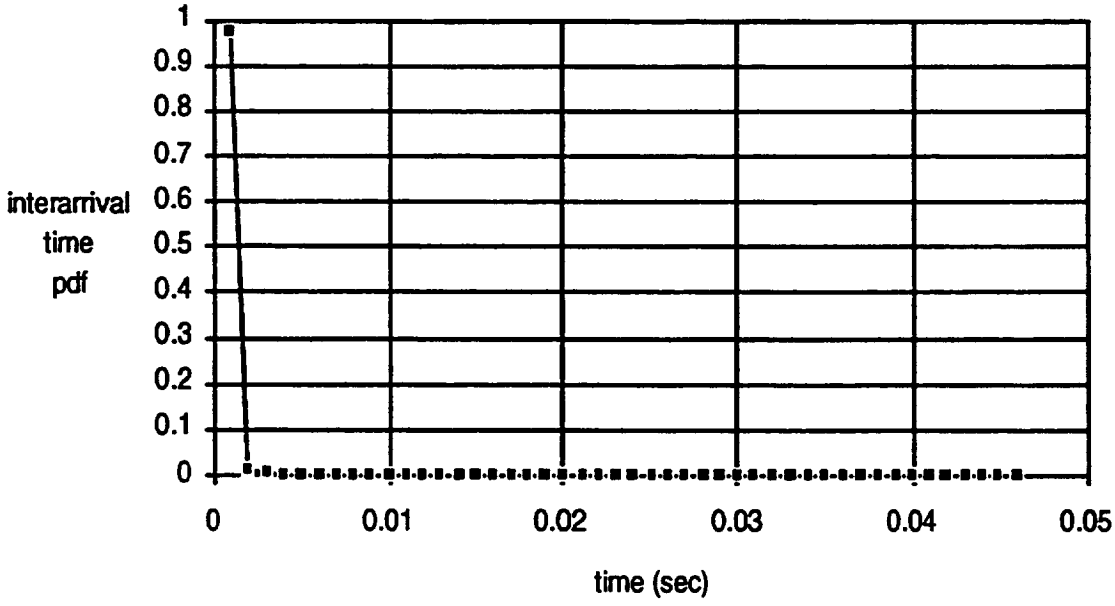

Figure 3-9 (b). Packet Interarrival Time pdf for Two-State MMPP

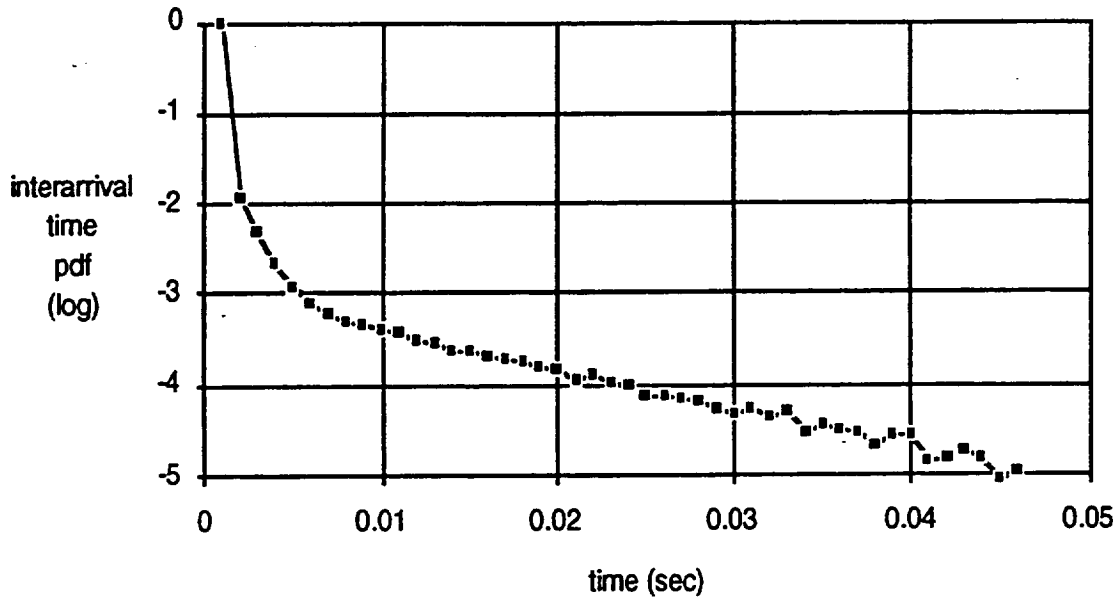

Figure 3-9 (c). Log Scale Packet Interarrival Time pdf for Two-State MMPP

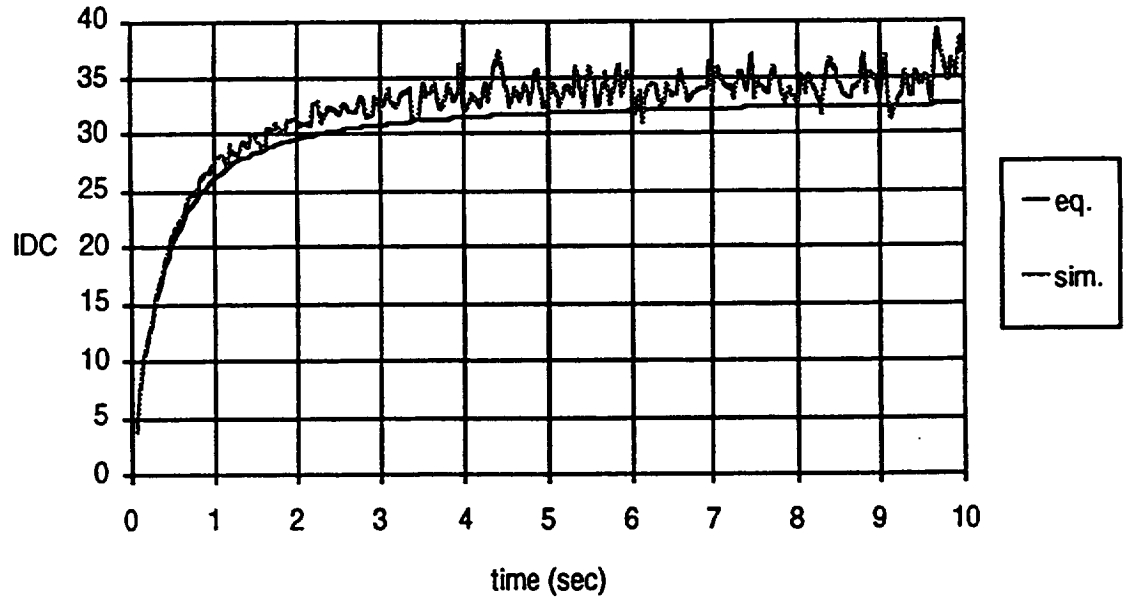

Figure 3-9 (d). Index of Dispersion for Counts (IDC) for Two-State MMPP

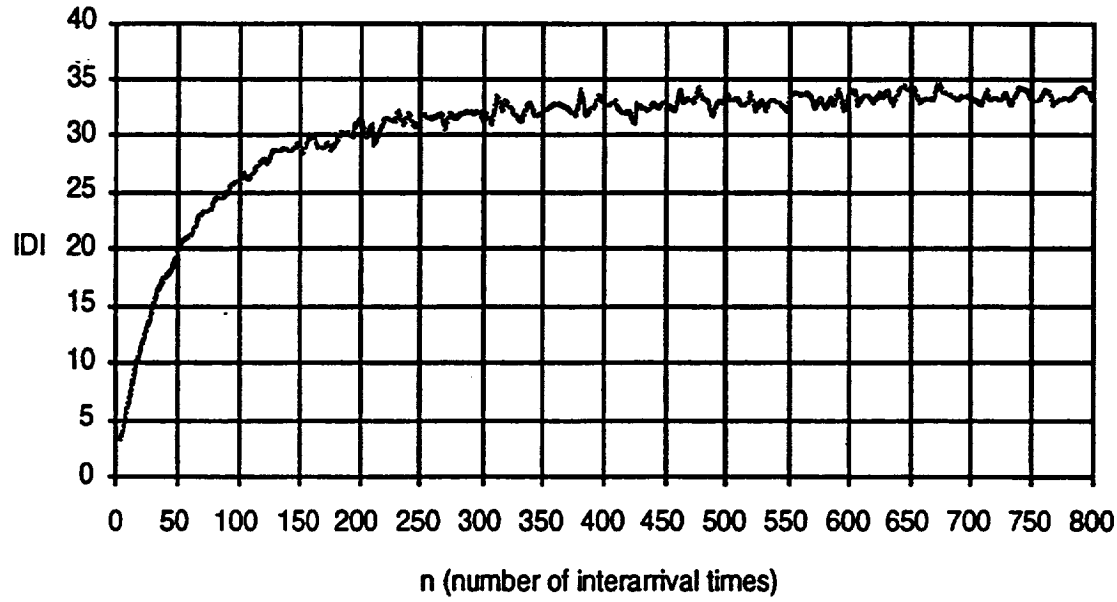

Figure 3-9 (e). Index of Dispersion for Intervals (IDI) for Two-State MMPP

3.2.8 Markov-Modulated Bernoulli Process

A discrete version of the MMPP, Markov-Modulated Bernoulli process (MMBP), was proposed in [3-18][3-28]. The durations of the state 1 and state 2 are assumed to be geometrically distributed. In each state, the packet arrival process is Bernoulli with loading ρ_i. The average offered loading is

$$\rho = \frac{\rho_1 E[T_1] + \rho_2 E[T_2]}{E[T_1] + E[T_2]} \tag{3-37}$$

The probability that state 1 consists of n packets is $p^{n-1}(1-p)$, where $n \geq 1$. State 1 consists of at least one cell. The $E[T_1]$ is $\frac{1}{1-p}$. The probability that state 2 consists of n cells is $q^{n-1}(1-q)$, where $n \geq 1$. State 2 consists of at least one cell. The mean off state duration $E[T_2]$ is $\frac{1}{1-q}$.

A simulation model is developed to analyze the characteristics of the two-state MMBP. The parameters used in the simulation are listed in Table 3-7. To completely specify the two-state MMBP, four parameters are required.

Table 3-7. Simulation Parameters for Two-State MMBP

PARAMETER	VALUE
link speed	2.048 Mbit/sec
packet slot time	0.001 sec
p	0.9996
ρ_1	0.98
q	0.996
ρ_1	0.1
mean link utilization	0.9

The simulated CV is 1.21. The simulation time is 2000 sec. The simulated interarrival time pdf, log scale interarrival time pdf, the IDC curve, and the IDI curve are shown in Figure 3-10.

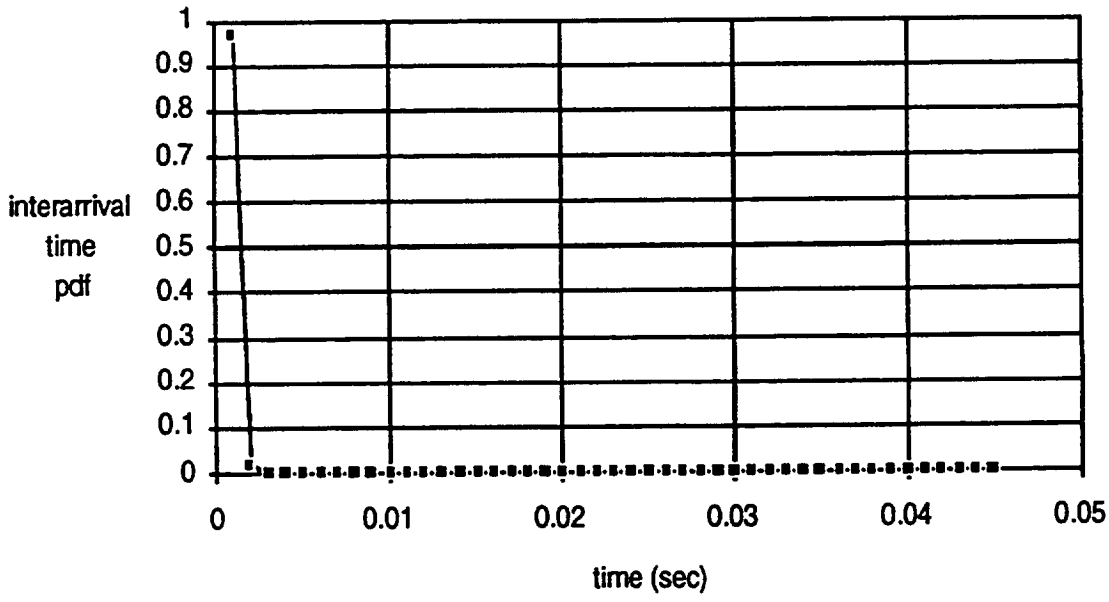

Figure 3-10 (a). Packet Interarrival Time pdf for Two-State MMBP

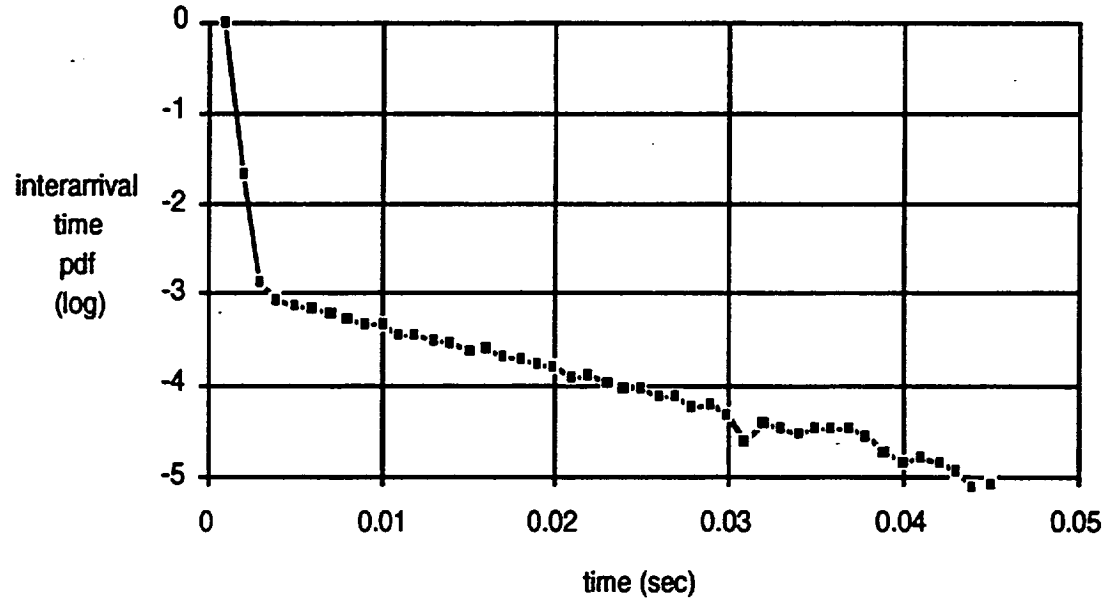

Figure 3-10 (b). Log Scale Packet Interarrival Time pdf for Two-State MMBP

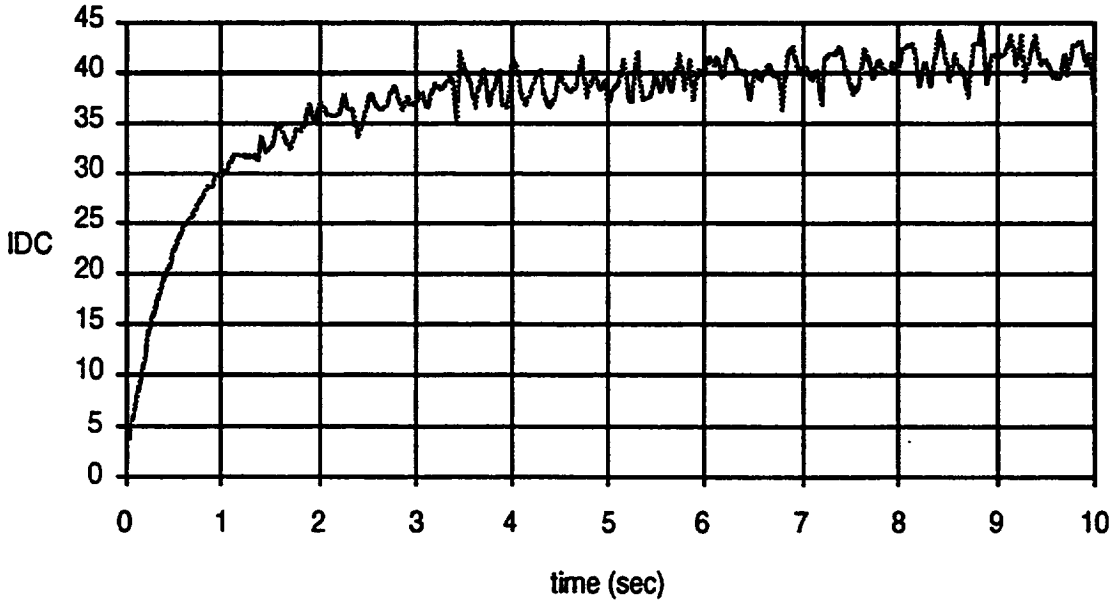

Figure 3-10 (c). Index of Dispersion for Counts (IDC) for Two-State MMBP

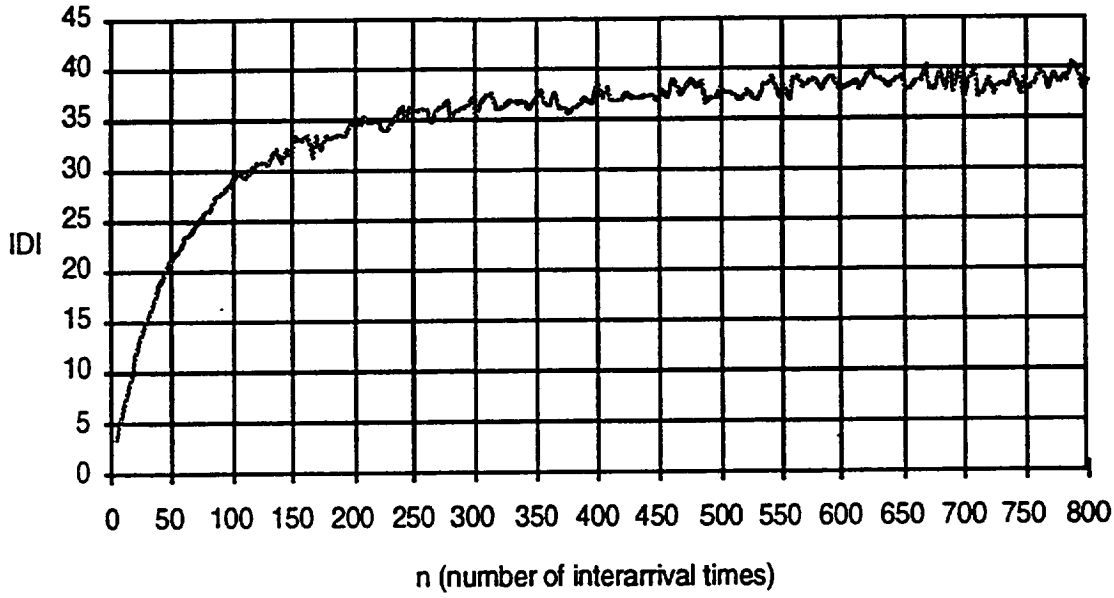

Figure 3-10 (d). Index of Dispersion for Intervals (IDI) for Two-State MMBP

3.2.9 Markov-Modulated Deterministic Process

A special case of the MMPP, Markov-Modulated deterministic process (MMDP), was proposed in [3-29]. The durations of the state 1 and state 2 follow exponential distribution. In each state, the packet arrival process is deterministic.

A simulation model is developed to analyze the characteristics of the two-state MMDP. The parameters used in the simulation are listed in Table 3-8. To completely specify the two-state MMDP, four parameters are required.

The simulated CV is 0.81. The simulation time is 2000 sec. The simulated interarrival time pdf, log scale interarrival time pdf, the IDC curve, and the IDI curve are shown in Figure 3-11.

Table 3-8. Simulation Parameters for Two-State MMDP

PARAMETER	VALUE
link speed	2.048 Mbit/sec
packet slot time	0.001 sec
mean bursty state duration ($E[T_1]$)	2.5 sec
bursty state deterministic interarrival time ($\frac{1}{\lambda_1}$)	0.00102 sec
mean normal state duration ($E[T_2]$)	0.25 sec
normal state deterministic interarrival time ($\frac{1}{\lambda_2}$)	0.01 sec
mean link utilization	0.9

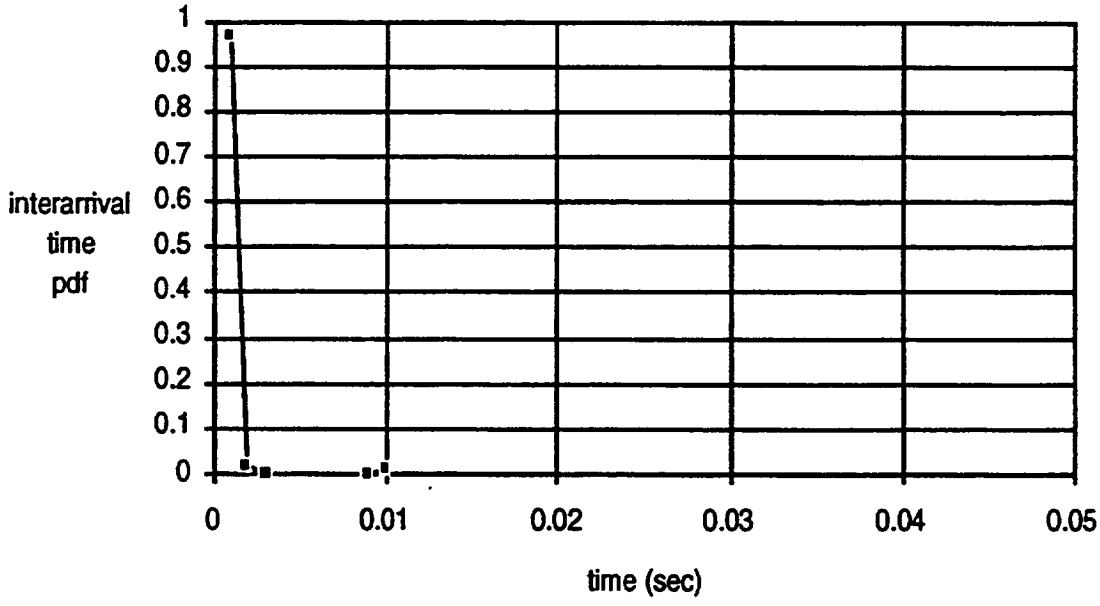

Figure 3-11 (a). Packet Interarrival Time pdf for Two-State MMDP

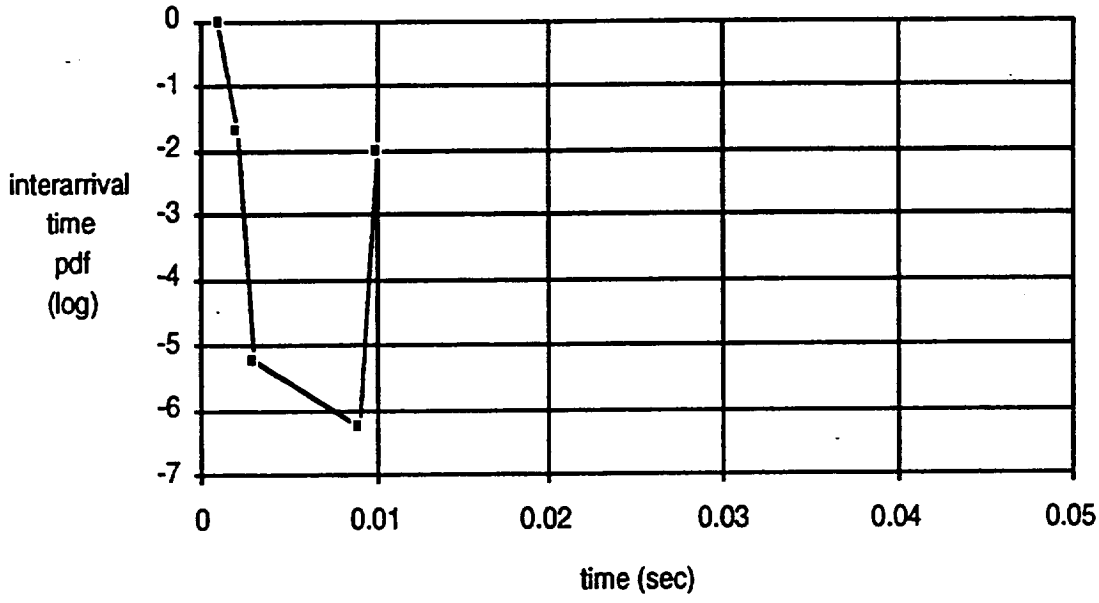

Figure 3-11 (b). Log Scale Packet Interarrival Time pdf for Two-State MMDP

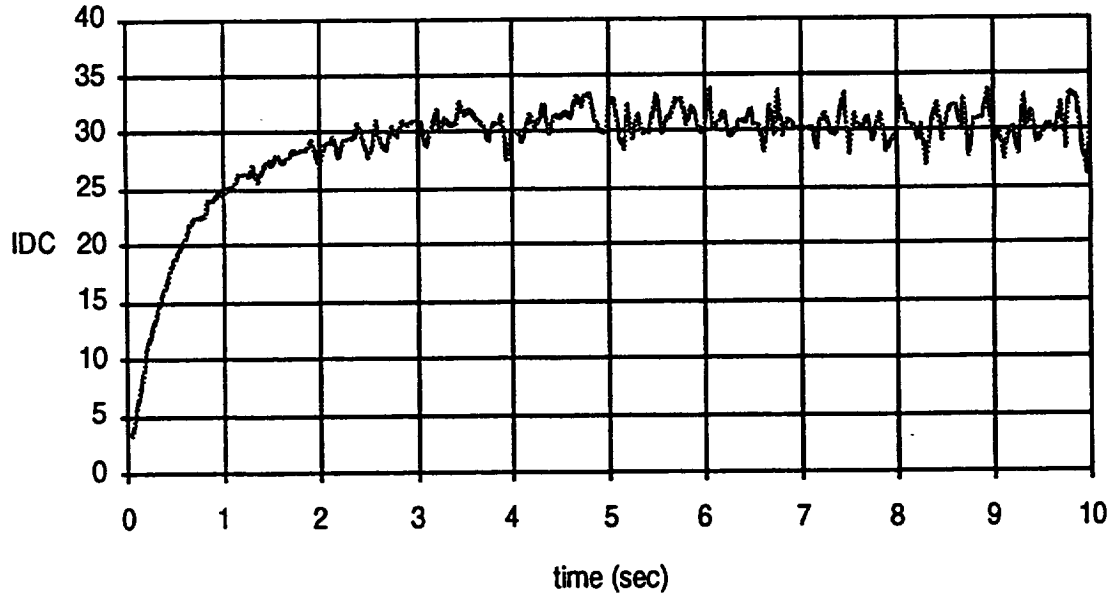

Figure 3-11 (c). Index of Dispersion for Counts (IDC) for Two-State MMDP

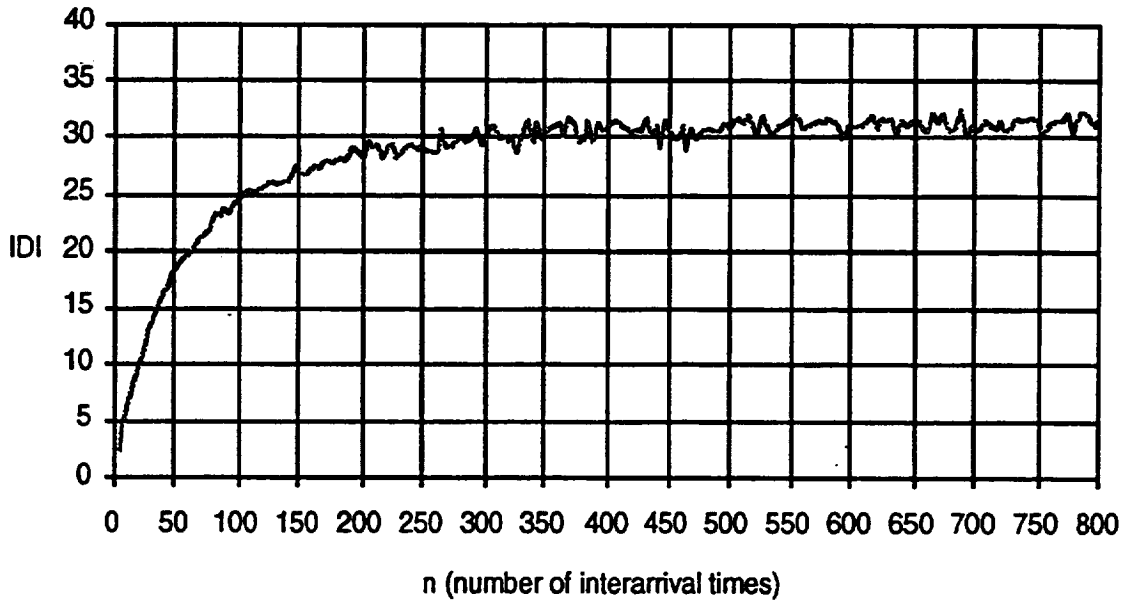

Figure 3-11 (d). Index of Dispersion for Intervals (IDI) for Two-State MMDP

3.2.10 On-Off or Interrupted Poisson Process

The on-off process is a special case of MMPP. The on-off process (or the interrupted Poisson process) has two states: on and off. The interarrival time of the on state follows an exponential distribution and the interarrival time of the off state follows another (independent) exponential distribution [3-5]. In other words, the interarrival time of bursts and burst size both follow an exponential distribution. During the on period, the packet arrival process is Poisson. There is no traffic during the off period. The cycles of the on-off process are a renewal process. Since the on state duration follows exponential distribution (a memoryless process), the packet interarrival times of the on-off process are also a renewal process. The MMPP with λ_1 and λ_2 is equivalent to the superposition of the on-off process with λ_1-λ_2 and the Poisson process with λ_2.

A special case of the on-off process is to let the packet interarrival time be deterministic during the on state. In other words, the source generates packets at a peak rate λ_{on} in on state. The on state duration and the off state duration will be Geometrically distributed, with respective mean values $E[T_{on}]$ and $E[T_{off}]$. Let P_{on} (or P_{off}) be the probability that the source is in the on state (or off state). Then

$$P_{on} = \frac{E[T_{on}]}{E[T_{on}]+E[T_{off}]} \quad \text{and} \quad P_{off} = \frac{E[T_{off}]}{E[T_{on}]+E[T_{off}]}. \tag{3-38}$$

This on-off process model is used in Reference 3-6 to model an ATM traffic source. Let the transition probability from the off state to the on state be α and that from the on

state to the off state be β (see Figure 3-12). The P_{on} and P_{off} can also be expressed using α and β as follows:

$$P_{off} = \frac{\beta}{\alpha+\beta} \text{ and } P_{on} = \frac{\alpha}{\alpha+\beta} \tag{3-39}$$

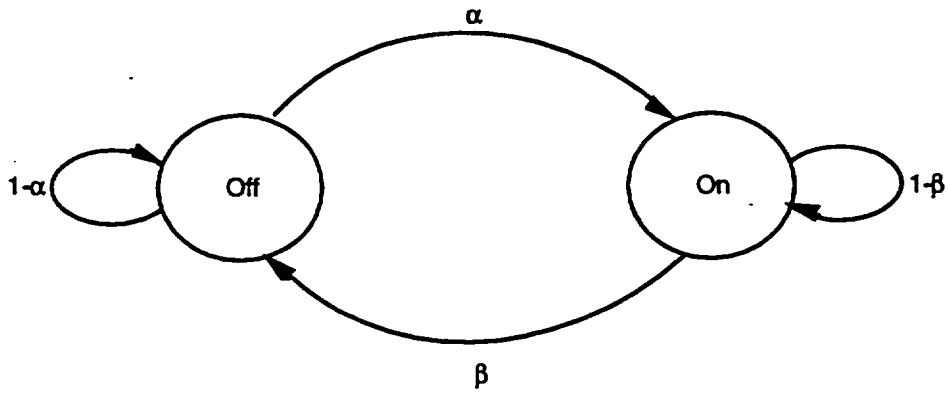

Figure 3-12. On-Off Process

The state transition probability α is set to $\frac{u}{b(1-u)}$ and β is set to $\frac{1}{b}$, where b is the average burst size (= $E[T_{on}]$) and u is equal to $\frac{E[\lambda]}{\lambda_{on}}$. Substitute α and β into Equation 3-39, P_{on} becomes $\frac{E[\lambda]}{\lambda_{on}}$. During the on (or bursty) state, cells are generated at the peak rate = $\frac{1}{T_{min}}$, where T_{min} is the cell interarrival time during the on state. The mean arrival rate is $E[\lambda] = \frac{\alpha}{T_{min}(\alpha+\beta)}$ as given in [3-4].

The on-off process used in Reference 3-19 is similar to that in Reference 3-6 except the on state generates cells according to the Bernoulli process.

In Reference 3-8, the periodicity of packet arrivals at the on state is investigated. In the on state, packets are generated at a constant rate $\frac{1}{T_{min}}$. T_{min} is the packet interarrival time at the on state and the average arrival rate is $E[\lambda]$, where $E[\lambda] = \frac{E[T_{on}]}{E[T_{on}]+E[T_{off}]} \frac{1}{T_{min}}$. However, the distribution of the on period may be different from that of the off period. The on period (or off period) is a sum of independent and identically distributed random variables. They are characterized by four parameters: mean on period, mean off period, variance of the on period and variance of the off period. A procedure was developed to select different distributions to match the measured mean

and variance of the on (or off) period. An 8 x 8 switch is used in the simulation. A typical traffic characteristic (destined to an output port) is given in Table 3-9.

Table 3-9. Typical Parameter Values for On-Off Process

λ_{on} (1/slot)	$E[T_{on}]$ (slot)	$E[T_{off}]$ (slot)	peak rate (Mbit/sec)	packet size (byte)
0.1	25	37	155.52	53

In Reference 3-9, the arrival process of ATM bursts (on states) is assumed to be Poisson. The burst size (number of cells) follows a Geometric distribution with a mean $\frac{1}{1-p}$. The probability that a burst consists of n cells is $p^{n-1}(1-p)$, where $n \geq 1$. The burst contains at least one cell. The spacing between any two cells in a given burst has a Geometric distribution. The probability that the interarrival time (spacing or separation) between two consecutive information cells is n cells is $\gamma^n(1-\gamma)$, where $n \geq 0$.

In Reference 3-10, both the on state duration and off state duration follow a Geometric distribution. The probability that a burst consists of n cells is $p^{n-1}(1-p)$, where $n \geq 1$. The burst contains at least one cell. The mean burst length $E[T_{on}]$ is $\frac{1}{1-p}$. During a burst, the packets are generated at the highest rate, i.e., the link speed. The probability that the off duration consists of n cells is $q^n(1-q)$, where $n \geq 0$. The mean off state duration $E[T_{off}]$ is $\frac{q}{1-q}$. The mean utilization is $\frac{E[T_{on}]}{E[T_{on}]+E[T_{off}]}$.

In Reference 3-16, the bursty ATM packet traffic source is modelled by an on-off process. The burst duration is random and the burst rate is deterministic. The off duration follows the exponential distribution.

In Reference 3-25, the on-off process model is used to model an ATM traffic source. Let the transition probability from the off state to the on state be α and that from the on state to the off state be β. The $E[T_{on}]$ is set to $\frac{1}{\beta}$ and $E[T_{off}]$ $\frac{1}{\alpha}$. At the on state, the source generates the packets according to the Poisson distribution. The off state duration follows the Geometric distribution. The characteristics of two sample ATM sources are listed in Table 3-10.

Table 3-10. Typical Parameter Values for On-Off Process

λ_{on} (1/slot)	$E[T_{on}]$ (slot)	$E[T_{off}]$ (slot)	peak rate (Mbit/sec)	packet size (byte)
1	10	100	155.52	53
1	100	1000	155.52	53

In Reference 3-12, the on-off model is used to model various traffic sources by properly choosing the peak rate λ_{on}, T_{on}, and P_{on}. For example, the constant bit rate source can be modelled by setting P_{on} = 1 and T_{on} = call duration. The voice source can be modelled by setting P_{on} in the range of [0.4-0.5] and T_{on} at about 1.2 sec.

In Reference 3-26, a superposition of N on-off sources is modeled by an N-state MMPP, where state k of the MMPP corresponds to k on sources and the arrival rate of k .

The periodic process is a special example of the on-off process. The on state duration T_{on} and the off state duration T_{off} are deterministic. During the on state, the packet interarrival time is also deterministic. For a constant bit rate process, the T_{off} becomes zero.

In Reference 3-14, a LAN traffic source is represented by an on-off process. The parameter values for the on-off process are listed in Table 3-11.

Table 3-11. Typical Parameter Values for On-Off Process

λ_{on} (1/sec)	E[T_{on}] (sec)	E[T_{off}] (sec)	peak rate (Mbit/sec)	packet size (byte)
2358.5	0.13	1.17	10	53

In Reference 3-23, the characteristics of data traffic for switched multi-megabit data service (SMDS) class 1 and class 3 services were reported. The parameter values are listed in Table 3-12.

Table 3-12. Typical Parameter Values for On-Off Process

λ_{on} (1/slot)	E[T_{on}] (slot)	E[T_{off}] (slot)	peak rate (Mbit/sec)	packet size (byte)
0.274	761	856	155.52	53

λ_{on} (1/slot)	E[T_{on}] (slot)	E[T_{off}] (slot)	peak rate (Mbit/sec)	packet size (byte)
0.274	761	5707	155.52	53

The on-off process was suggested to model packetized voice [3-13], still picture [3-27], and interactive data service.

A simulation model is developed to analyze the characteristics of the on-off process. The parameters used in the simulation are listed in Table 3-13. To completely specify the on-off process, three parameters are required.

Table 3-13. Simulation Parameters for On-Off Process

PARAMETER	VALUE
link speed	2.048 Mbit/sec
packet slot time	0.001 sec
mean on state duration ($E[T_{on}]$)	2.5 sec
deterministic on state interarrival time ($\frac{1}{\lambda_{on}}$)	0.00101 sec
mean off state duration ($E[T_{off}]$)	0.25 sec
mean link utilization	0.9

The simulated CV is 6.44. The simulation time is 2000 sec. The simulated interarrival time pdf, log scale interarrival time pdf, the IDC curve, and the IDI curve are shown in Figure 3-13.

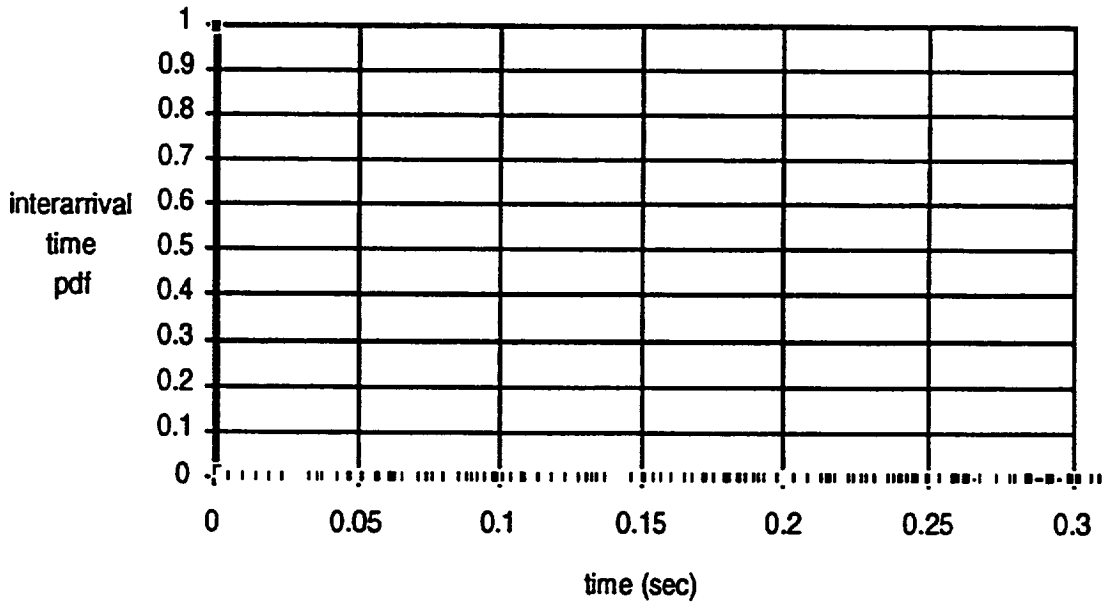

Figure 3-13 (a). Packet Interarrival Time pdf for On-Off Process

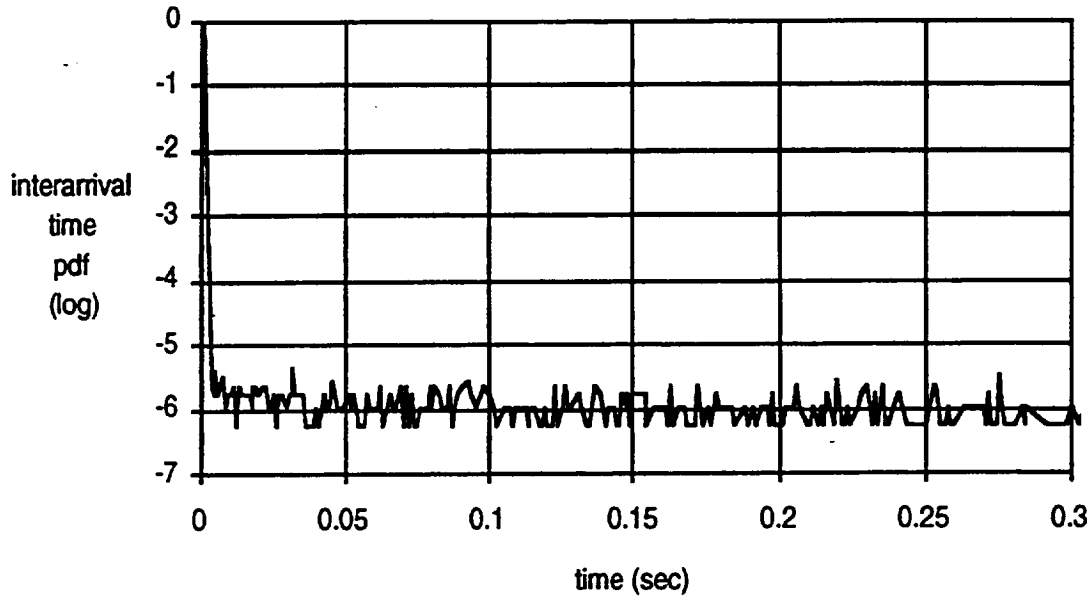

Figure 3-13 (b). Log Scale Packet Interarrival Time pdf for On-Off Process

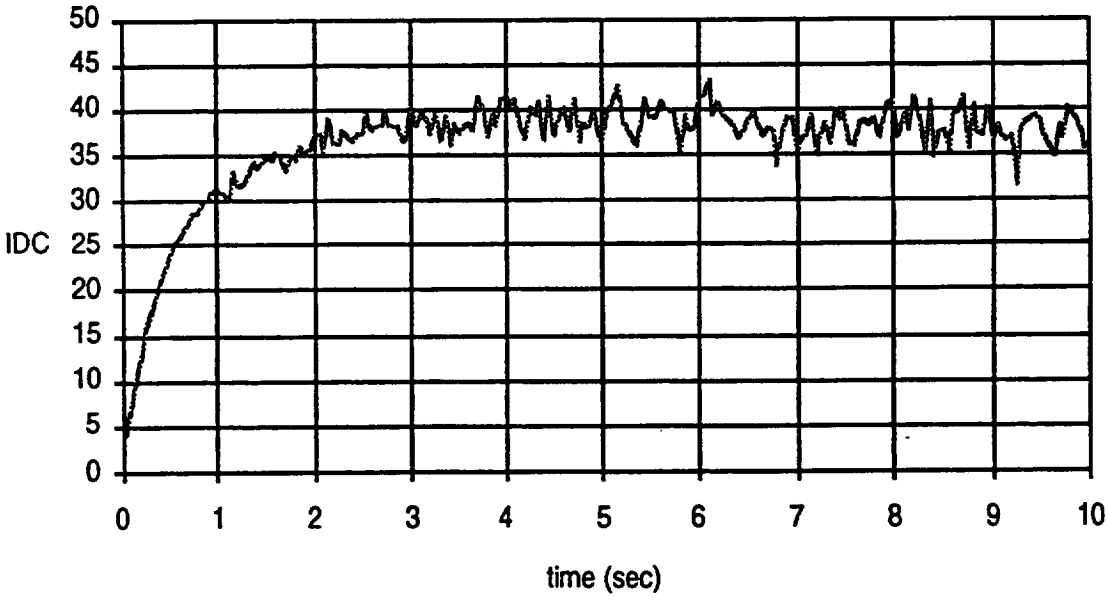

Figure 3-13 (c). Index of Dispersion for Counts (IDC) for On-Off Process

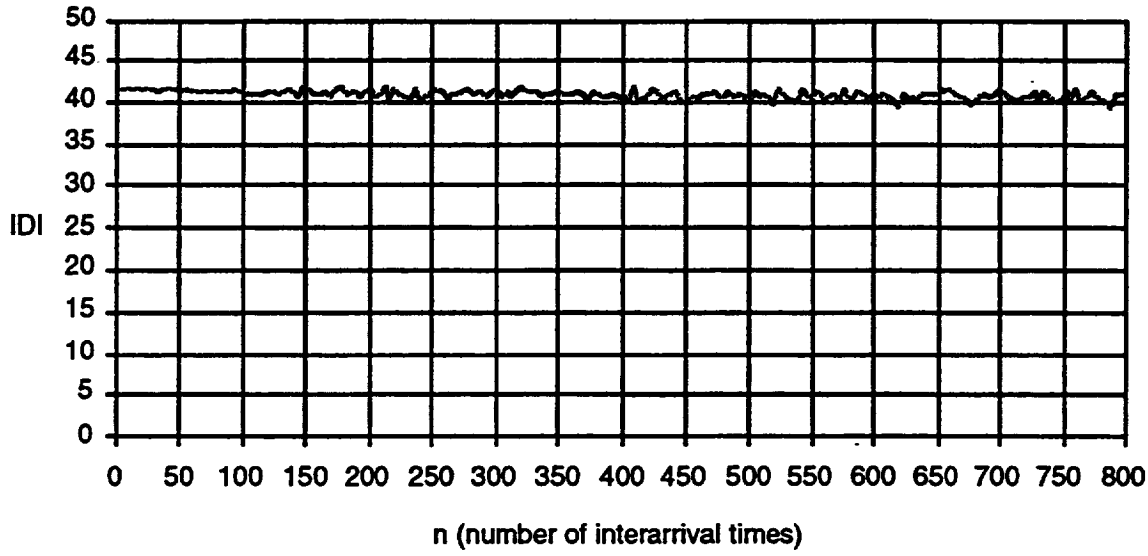

Figure 3-13 (d). Index of Dispersion for Intervals (IDI) for On-Off Process

3.2.11 Packet Train

The packet train uses a cluster of packets traveling in both directions between a node pair to model the packet arrivals in a ring LAN [2-2]. There is an inter-packet gap within a cluster. If the interarrival time of the packets is less then the gap, they are considered to be on the same train. If not, the next packet will be the start of a new train. The concept of the packet train is very similar to the on-off model except the packet train also captures the dependences on the destination.

3.3 Voice Source Models

For easy analysis, the voice sources in the DDPS satellite network are assumed to be homogeneous. The call arrival process is Poisson and the call duration follows an exponential distribution. The amount of capacity allocated for voice connections depends on the QOS requirement and the statistical multiplexing gain. The statistical multiplexing gain is about 2 if more than 24 voice sources are multiplexed [3-21]. A voice call is characterized by alternating talkspurts and silences. It is reasonable to assume that the interval of a talkspurt and that of the successive silence are independent. Therefore, the packet arrival process for a voice call can be modelled by a renewal process, i.e., the packet arrival times are independent and identically distributed. The on-off process described above has been suggested to model a single voice source. The characteristics of a voice source is given in Table 3-14 [3-12] [3-24].

Table 3-14. Typical Parameter Values for Voice Source

λ_{on} (1/sec)	E[T_{on}] (sec)	E[T_{off}] (sec)	peak rate (kbit/sec)	packet size (byte)
75.5	1.0-1.2	1.4	64 or 32	53
62.5	0.352	0.65	32	64

The accuracy of the on-off model is quite high if the number of voice sources is larger than 25 [3-17]. Let the transition probability from the off state to the on state be α and that from the on state to the off state be β.

The probability that there are k number of on input lines out of N voice sources is

$$P_k = \binom{N}{k} P_{on}{}^k P_{off}{}^{N-k}, \text{ where } k \le N.$$

$$P_k = \binom{N}{k} \left(\frac{\alpha}{\alpha+\beta}\right)^k \left(\frac{\beta}{\alpha+\beta}\right)^{N-k} \tag{3-40}$$

where $k \le N$. This model is called the Engest distribution.

P_k can also be derived using the birth-death process (see Figure 3-14). The arrival rate $\gamma_k = (N-k)\,\alpha$ and the death rate $\mu_k = n\beta$. In state k, since there are k calls are in talkspurt, the total arrival rate is $k\,\lambda$.

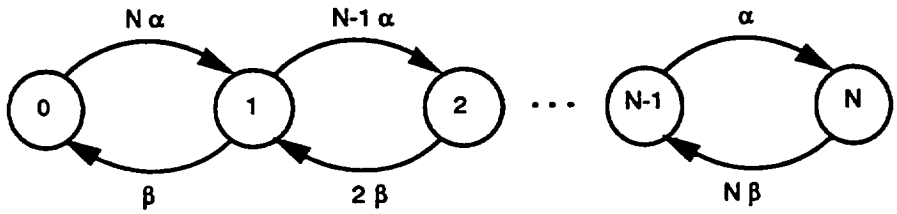

Figure 3-14. Birth-Death Process for Arrival Rate

The equilibrium state solution of the birth-death process is given as

$$P_k = \frac{\gamma_0\,\gamma_1 \cdots \gamma_{k-1}}{\mu_1\,\mu_2 \cdots \mu_k} P_0$$

$$= \left(\frac{\alpha}{\beta}\right)^k \frac{N\,(N-1) \ldots (N-k+1)}{1\,2 \ldots k} P_0$$

$$= \left(\frac{\alpha}{\beta}\right)^k \binom{N}{k} P_0 \tag{3-41}$$

By conservation of probability, we have

$$P_0 = \frac{1}{\sum_{i=0}^{N} (\frac{\alpha}{\beta})^i \binom{N}{i}}$$

$$P_k = \frac{(\frac{\alpha}{\beta})^k \binom{N}{k}}{\sum_{i=0}^{N} (\frac{\alpha}{\beta})^i \binom{N}{i}} = \frac{(\frac{\alpha}{\beta})^k \binom{N}{k}}{(1+\frac{\alpha}{\beta})^N} = \binom{N}{k} (\frac{\alpha}{\alpha+\beta})^k (\frac{\beta}{\alpha+\beta})^{N-k} \qquad (3\text{-}42)$$

The arrival process in a talkspurt can be either deterministic or a Poisson process.

The superposition of a large number of voice sources can be approximated by a Poisson process only when the traffic intensity is low [3-24]. A better model for the aggregate voice traffic is to use the MMPP [3-13]. The determination of the four parameters for the MMPP are described in Section 3.5. From the simulation results (not shown here), the multiplexed voice traffic can not be represented by an on-off process.

3.4 Multiplexing of Data and Voice

In general, specifying queueing models for the multiplexed traffic of data and voice is very difficult. The law of a large number states that the normalized sum of a large number of random variables is close to the mean with a high probability. Applying the result to the packet switched network, the outcome of multiplexing many independent packet streams is that statistical fluctuations of packet streams are smoothed out and the aggregate traffic volume is close to the mean traffic volume with a high probability. Although this statement is helpful for bandwidth allocation, it does not reveal any information about the burstiness and correlation between packets. A simple queueing model such the Poisson process is not capable of modeling the burstiness of the traffic resulted from multiplexing different traffic types. The correlation of packets observed in the multiplexed traffic makes the renewal process such as the on-off model questionable [3-7]. However, if queueing models are so complex that analytical approach becomes intractable, then the value of the traffic model becomes insignificant. A compromise between modeling accuracy and analytical tractability must be exercised when choosing a source queueing model for the traffic generator.

In reality, the traffic intensity depends on the time of the day (such as busy hours). This implies parameters are time varying. For example, the state durations and the arrival rates for each state for the MMPP can become time varying.

Four approaches are identified to model the multiplexed traffic source. The first approach is to use one queueing model to represent the superposition of different traffic types. The queueing model suggested is the two-state MMPP since it is the most generalized and widely used model for the bursty traffic and superposition of packetized voice and data traffic [3-13]. MMPP is capable of capturing traffic burstiness and

correlation and is simple for analysis. The two-state MMPP was used to fit the measured data from a workstation and the fitting result was encouraging [3-3]. Algorithms to fit the measurement data to the two-state MMPP are described later.

The second approach is to use the MMPP to model the multiplexed voice sources and the data sources are modeled by a Poisson process. Remember that the superposition of the MMPP (with λ_1 and λ_2) with an independent Poisson process (with λ_3) also results in an MMPP (with $\lambda_1 + \lambda_3$ and $\lambda_2 + \lambda_3$). This suggested that the aggregate traffic pattern depends on $\lambda_1 + \lambda_3$ and $\lambda_2 + \lambda_3$, not their individual value. Therefore, the aggregate traffic pattern is the same for 30% voice, 70% data and 70% voice, 30% data. This may not be accurate.

The third approach is to use one MMPP to model the multiplexed voice traffic and another MMPP to model the data traffic. This approach allows the user to be able to characterize the voice traffic and data traffic individually. However, a statistical multiplexer is required to multiplex the output of these sources into a high-speed TDM packet stream.

The fourth approach is to allow the user to have the capability of creating multiple (different) source queueing models to represent the multiplexed traffic. A statistical multiplexer is required to multiplex the output of these sources into a high-speed TDM packet stream. Note that when the packets pass through the statistical multiplexer, the packets suffer queueing delay. Clumping and dispersion are two effects of packet delay variation. Clumping refers to that the instaneous peak rate is higher than the original peak rate. Dispersion refers to that the interarrival time is larger then that of the original process. For delay-sensitive data, the packet delay variation has to be compensated at the end-user.

The conceptual configuration of these four approaches are illustrated in Figure 3-15. The first approach is recommended.

A summary, which includes the applications, the capability, and the required parameters for each source queueing model, is listed in Table 3-15.

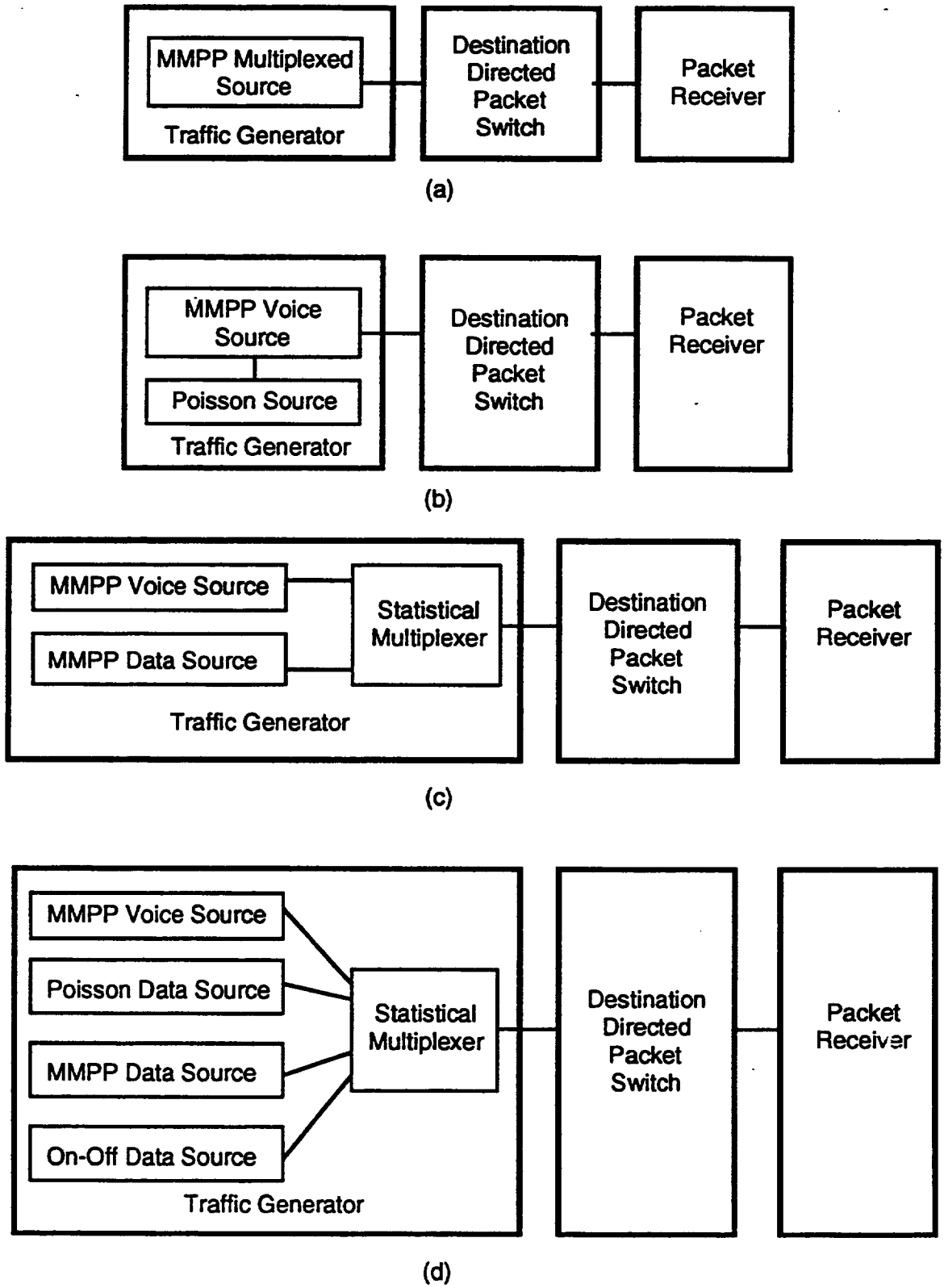

Figure 3-15. Four Approaches to Model Multiplexed Traffic

Table 3-15. A Summary of Different Source Queueing Models

Model	Applications	B	C	parameters
Poisson	random traffic, superposition of a large number of low speed voice and data, background traffic.	no	no	arrival rate
Geometric	same as Poisson	no	no	mean util.
Batch Poisson	bursty data	yes	no	arrival rate, batch size
Hyperexponential	bursty data	yes	no	arrival rates (2), coefficient
MMPP	superposition of packetized voice and data, ATM source	yes	yes	state durations (2), arrival rates (2)
On-Off	superposition of packetized voice and data, ATM source	yes	yes	state durations (2), arrival rate

B: burstiness and C: correlation

3.5 Fitting Algorithms of the Measurement Data to a Traffic Source Model

After measurements are performed on real traffic sources, it is important to choose a queuing model and its associated parameters such that the model can be used to represent the real traffic pattern, and subsequently used in the traffic generator of the testbed. Although the parameters for voice traffic are well documented, it is not easy to choose the parameters for data traffic. This subsection presents the algorithms of fitting the measured traffic characteristics from real traffic sources to a queueing source model. The two-state MMPP is the most generalized and widely used traffic model to represent the bursty traffic and the superposition of packetized voice and data traffic. The two-state MMPP is used as the representative for discussion. Many algorithms have been proposed to fit the measurement data to the queueing model. Most algorithms analyze the observed data offline; only one real-time traffic characterization algorithm is available [3-1].

3.5.1 Offline Algorithms

The fitting algorithm described in Reference 3-3 computes the first moments and the second moments of the state durations and the associated interarrival times using four equations.

$$a = E[t] = 1/E[\lambda] = \frac{E[T_1] + E[T_2]}{\lambda_1 E[T_1] + \lambda_2 E[T_2]} \tag{3-43}$$

$$b + 1 = L_\infty + 1 = 1 + \frac{2(E[T_1]E[T_2])^2 (\lambda_1 - \lambda_2)^2}{(E[T_1]+E[T_2])^2 (E[T_1]\lambda_1 + E[T_2]\lambda_2)} \tag{3-44}$$

$$c = r = \frac{1}{E[T_1]} + \frac{1}{E[T_2]} \tag{3-45}$$

$$d = E[(\lambda)^2] = \frac{(\lambda_1)^2 E[T_1] + (\lambda_2)^2 E[T_2]}{E[T_1] + E[T_2]} \tag{3-46}$$

Use a, b, c, and λ_2 to express $E[T_1]$, $E[T_2]$, and λ_1. We obtain

$$E[T_1] = \frac{abc + 2(1-a\lambda_2)^2}{abc^2} \tag{3-47}$$

$$E[T_2] = \frac{abc + 2(1-a\lambda_2)^2}{2c(1-a\lambda_2)^2} \tag{3-48}$$

$$\lambda_1 = \frac{abc + 2(1-a\lambda_2)}{2a(1-a\lambda_2)} \tag{3-49}$$

The values of a, b, and d can be obtained by measurements. However, the value of c must be obtained by numerically solving Equation 3-35 with measured I_t at a give time t and I_∞ [3-3]. Substitute Equations 3-47, 3-48, and 3-49 to Equation 3-46 for d. The value for λ_2 can be obtained by finding the roots of a polynomial of degree 2. In general, there are two solutions for λ_2. The Laguerre's method can be applied to numerically find the roots of the polynomial. The C program for the Laguerre's method can be found in [3-31]. After λ_2 is obtained, the values for $E[T_1]$, $E[T_2]$, and λ_1 can be easily calculated.

These estimated parameters are used to plot a theoretical IDC curve. The theoretical IDC curve from the estimated parameters is compared with the measured IDC curve for accuracy. If the approximated parameters are not accurate enough, another value of c is used to repeat the same procedure.

The fitting algorithm described in Reference 3-13 is similar to that in Reference 3-3. The algorithm also uses four equations to obtain the moments of state durations and their interarrival times. The first three equations are the same as Equations 3-43, 3-44, and 3-45. The third moment of t_2 is used as Equation 3-46.

Reference 3-8 describes a procedure of selecting distributions for the on state and off state durations of the on-off process. The selected distribution only matches the first and the second moments of the observed on or off durations. The first moment uses the measured mean E[T] and the second moment uses the squared coefficient of variance (CV^2). When $E[T] > 2$ and $CV^2 > 1 - \frac{1}{E[T]}$, the (on or off) duration is modelled as a mixture of two modified Geometric distributions. The pdf of the (on or off) duration is defined as

$$P(n) = \alpha(1-p_1)(p_1)^{n-1} + (1-\alpha)(1-p_2)(p_2)^{n-1}, \quad n \geq 1. \tag{3-50}$$

The mean for the pdf is $\frac{\alpha}{1-p_1} + \frac{1-\alpha}{1-P_2}$. The proper choices of α, p_1 and p_2 are

$$\alpha = 0.5\,(1 + (\frac{(CV^2-1)E[T] + 1}{(CV^2+1)E[T] + 1})^{0.5}) \tag{3-51}$$

$$p_1 = (E[T]-2\alpha)/E[T], \tag{3-52}$$

$$p_2 = (E[T] - 2(1-\alpha))/E[T]. \tag{3-53}$$

When $\frac{1}{E[T]-1} < CV^2 < 1$, the (on or off) duration is modelled as a summation of N independent and identically distributed modified geometric random variables. Let the summation be X.

$$X = 1 + x_1 + x_2 + \ldots + x_N. \tag{3-54}$$

The pdf of each random variable is

$$P(n) = (1-p)\,(p)^n\,,\ n \geq 0. \tag{3-55}$$

The mean is

$$E[X] = 1 + \frac{Np}{1-p}. \tag{3-56}$$

The proper choices of N and p are

$$N = \frac{E[T]-1}{((E[T]-1)CV^2 - 1)} \tag{3-57}$$

$$p = \frac{E[T]-1}{N+E[T]-1}. \tag{3-58}$$

Note that CV^2 is an approximation since N must be an integer.

3.5.2 Online Algorithms

The following describes several simple fitting algorithms for the MMPP (or its variations such as MMBP, MMDP, and on-off process). The on-line fitting algorithms are useful for real-time network management and traffic management. The key is that a cutoff point for packet interarrival time, which delimits state 1 and state 2, must be found for a two-state MMPP. Normally, the traffic intensity of one state is higher than that of the other state. Denote these two states as bursty state and normal state. After the cutoff point is obtained, the collected traffic profile can be viewed as a profile which alternates between the burst state and the normal state. To illustrate the usefulness of these procedures, experiments are performed. A two-state MMPP with known parameters is used as the traffic source. The traffic source generates information packets according to the specified queueing model. A packet receiver collects the statistics. These algorithms are implemented at the packet receiver to estimate the shape or the moments of distribution.

The first algorithm is to use the histogram approach. The histogram is useful to identify the shape of the distributions (for interarrival times and state durations) and to estimate the first two moments. A histogram is constructed by

1. Dividing the range of data into equal-size bins. For example, the bin size for the packet interarrival times is the packet slot time.

2. Determining the frequency of occurrence for each bin.

The histogram corresponds to a pdf. By plotting the histogram of the packet interarrival time of the MMPP, λ_1 and λ_2 may be determined. The main goal is to find a cutoff point so that the histogram can be divided into two histograms, one for the bursty state and the other for the normal state. The cutoff point is hard to visualize in the linear histogram; however, the log scale histogram can clearly illustrate the composition of two histograms. An example is shown in Figures 3-16 and 3-17. The simulation parameters of the MMPP are the same as those in Section 3.2.8. Three interarrival time linear scale histogram curves are shown in Figure 3-16: the bursty state, the normal state, and the combined one. The bin size of the histogram is the packet slot time (0.001 sec). The interarrival time log scale histograms are shown in Figure 3-17.

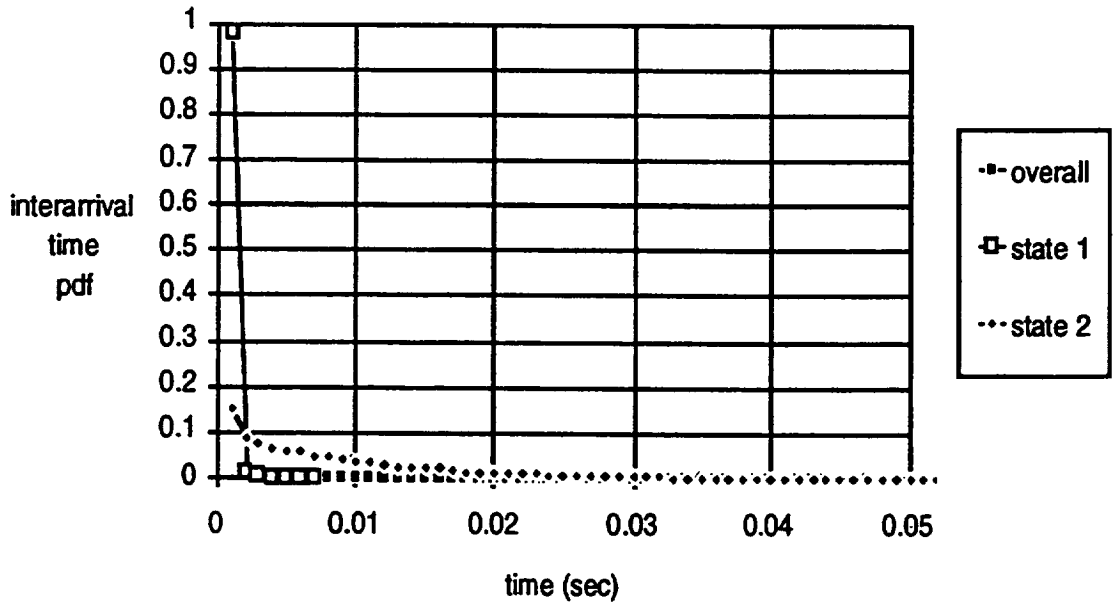

Figure 3-16. Packet Interarrival Time pdf for Original Two-State MMPP

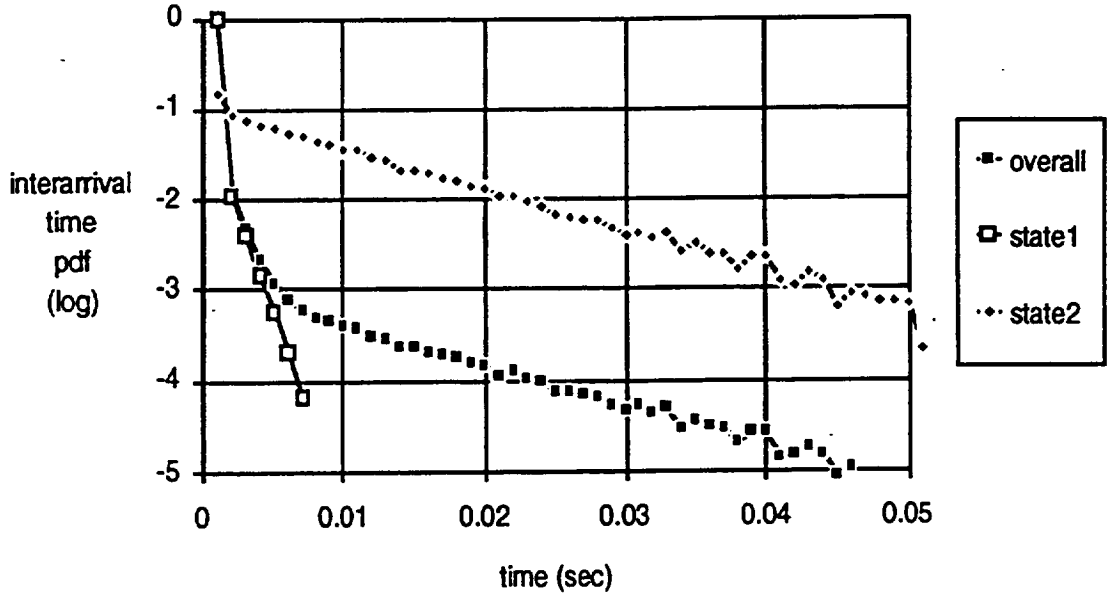

Figure 3-17. Log Scale Packet Interarrival Time pdf for Original Two-State MMPP

The second algorithm, which is a simple, real time traffic model fitting algorithm, was proposed in Reference 3-1. The advantages of real time traffic analysis are: the real time results are useful for network management, traffic management (such as resource allocation), and admission control. In this algorithm, samples of packet interarrival time (t_i) are collected. The algorithm compares the current sample and the previous sample. By performing this comparison, the algorithm makes a decision whether the current sample belongs to the previous state or the current sample is the beginning of a new state. This algorithm is referred as "time comparison" algorithm. This type of decision is referred as hypothesis testing. There are two hypotheses -- H0: the two samples are in the same state and H1: the two samples are in different state. A constant C is selected as a threshold. If $\frac{t_i}{t_{i+1}} \geq C(t)$ or $\frac{t_{i+1}}{t_i} \geq C(t)$, then H0 is false. Otherwise, H0 is true. The value of C(t) is hard to choose because it can not favor H0 nor H1. A balance point must be obtained. The optimum value of C(t) would be close to $\frac{\lambda_1}{\lambda_2}$, if $\lambda 1$ and $\lambda 2$ are known in advance. To improve the accuracy of the time comparison algorithm, it is suggested in [3-1] that the previous sample is substituted with an average mean of interarrival times for the state. The algorithm works well when the traffic is very bursty, i.e., $\frac{\lambda_1}{\lambda_2} >> 1$. In this case, the value of C(t) can be set very high (e.g. 10) and the decision error becomes small. This seems to be reasonable. If the traffic is not bursty, then the switch performance degradation due to burstiness is small; therefore, the accuracy of the fitting algorithm of finding the parameter values of the MMPP has no major impact to the switch performance. Only when the traffic is very bursty, an accurate fitting algorithm is required to generate the parameter values of the MMPP such that the switch performance can be evaluated in a precise manner.

The time comparison algorithm not only can obtain the parameter values for the MMPP, it but also can obtain the parameters values for the on-off process. Assume initially the state is at the on state. During the on state, since packet arrivals are deterministic, $\frac{t_i}{t_{i+1}} = 1$. When $\frac{t_i}{t_{i+1}} \neq 1$, it means that t_i is associated with the last packet of the previous on state and t_{i+1} is associated with the first packet of a new on state. If this happens, the state will transit to the other state. However, since $\frac{t_{i+1}}{t_{i+2}}$ will be equal to 1, the state will transit back to the on state. The estimated arrival time for the off state is actually the off state mean duration.

Applying the same principle as in Reference 3-1, a new real-time traffic model fitting algorithm is proposed. Instead of using the packet arrival times as samples, the number of arrivals (N_i) is used as samples for comparison. This algorithm is referred as the "count comparison" algorithm. The number of packet arrivals in an interval t (called bin) is compared with that in the previous bin. There are two hypotheses -- H0: the two samples are in the same state and H1: the two samples are in different state. A constant C(N) is selected as a threshold. If $\frac{N_i}{N_{i+1}} \geq C(N)$ or $\frac{N_{i+1}}{N_i} \geq C(N)$, then H0 is false. Otherwise, H0 is true. The old sample is an average mean of all the samples in the same state. Clearly the bin size must be small enough to catch the states with short durations. The accuracy of this algorithm depends on the values of bin and C(N). An experiment using the algorithm to estimate the original traffic characteristics is performed. The parameter values used for the count comparison algorithm are listed in Table 3-16.

Table 3-16. Parameter Values used for Count Comparison Algorithm

bin size	C(N)
0.05 sec	3.5

The reconstructed traffic characteristics for different source models are shown in Tables 3-17 to 3-20.

Table 3-17. Characteristics of Original MMPP and Reconstructed Traffic Patterns

	Original	Reconstructed
$E[T_1]$	2.5 sec	2.576 sec
$(\frac{1}{\lambda_1})$	0.00102 sec	0.001038 sec
$E[T_2]$	0.25 sec	0.243 sec
$(\frac{1}{\lambda_2})$	0.01 sec	0.00857 sec

Table 3-18. Characteristics of Original MMBP and Reconstructed Traffic Patterns

	Original	Reconstructed
$E[T_1]$	2.5 sec	2.78 sec
$(\frac{1}{\lambda_1})$	0.00102 sec	0.001033 sec
$E[T_2]$	0.25 sec	0.267 sec
$(\frac{1}{\lambda_2})$	0.01 sec	0.00891 sec

Table 3-19. Characteristics of Original MMDP and Reconstructed Traffic Patterns

	Original	Reconstructed
$E[T_1]$	2.5 sec	3.25 sec
$(\frac{1}{\lambda_1})$	0.00102 sec	0.001029 sec
$E[T_2]$	0.25 sec	0.289 sec
$(\frac{1}{\lambda_2})$	0.01 sec	0.0089 sec

Table 3-20. Characteristics of Original On-Off Process and Reconstructed Traffic Patterns

	Original	Reconstructed
$E[T_1]$	2.5 sec	3.12 sec
$(\frac{1}{\lambda_1})$	0.00102 sec	0.001023 sec
$E[T_2]$	0.25 sec	0.26 sec
$(\frac{1}{\lambda_2})$	x	0.12789 sec

The above two algorithms suffer the same problem: once a false transition is made, the errors are accumulated. In other words, when a false transition is made, the false sample statistics is used to compute a new average for the interarrival time of the state. The accuracy of the average becomes less and less when more and more false samples are used. A new algorithm is proposed to overcome this problem. The major discovery is that it is not necessary to make a precise decision for every sample. Uncertain samples can be discarded and no state transition is made; as a result, the false transitions and error accumulation can be largely eliminated. The algorithm follows that of the TDM synchronization or ATM cell self delineation scheme. Acquiring and maintaining the characteristics of the state of the packet arrival process is analogous to acquiring and maintaining the synchronization of the TDM frames. The state is either

at state 1 or state 2 and the synchronizer is either in sync or out of sync. The state transition decision mechanism is the same as that in Reference 3-1.

The first new time comparison algorithm is there are two states: normal and bursty. Let the initial state be in the normal state. This algorithm is termed as the "time comparison algorithm II". In order to transit from the normal state to the bursty state, the number of consecutive failed hypotheses for the two samples being in the same state must be greater than or equal to T_{nb}. When the number of failed hypotheses is less than T_{nb}, the new samples under test are discarded. In order to transit from the bursty state to the normal state, the number of consecutive failed hypotheses must be greater than or equal to T_{bn}. When the number of failed hypotheses is less than T_{bn}, the new samples are discarded. The state diagram is shown in Figure 3-18.

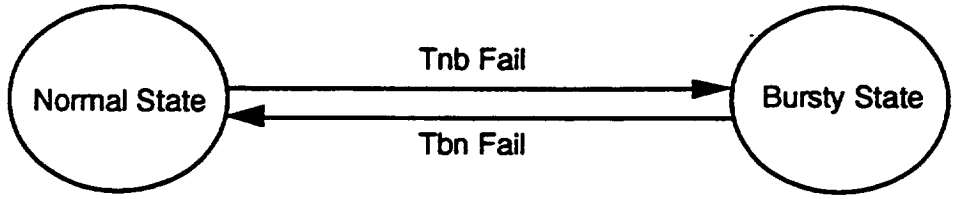

Figure 3-18. State Diagram of Time Comparison Algorithm II

The reconstructed traffic characteristics for different source models are shown in Tables 3-21 to 3-24.

Table 3-21 (a). Parameter Values used for Time Comparison Algorithm II

C(T)	T_{nb} (T_{bn})
6	2

Table 3-21 (b). Characteristics of Original MMPP and Reconstructed Traffic Patterns

	Original	Reconstructed
$E[T_1]$	2.5 sec	2.09 sec
$(\frac{1}{\lambda_1})$	0.00102 sec	0.001029 sec
$E[T_2]$	0.25 sec	0.157 sec
$(\frac{1}{\lambda_2})$	0.01 sec	0.01156 sec

Table 3-22 (a). Parameter Values used for Time Comparison Algorithm II

C(T)	T_{nb} (T_{bn})
6	2

Table 3-22 (b). Characteristics of Original MMBP and Reconstructed Traffic Patterns

	Original	Reconstructed
$E[T_1]$	2.5 sec	2.419 sec
$(\frac{1}{\lambda_1})$	0.00102 sec	0.001023 sec
$E[T_2]$	0.25 sec	0.204 sec
$(\frac{1}{\lambda_2})$	0.01 sec	0.0109 sec

Table 3-23 (a). Parameter Values used for Time Comparison Algorithm II

C(T)	T_{nb} (T_{bn})
6	1

Table 3-23 (b). Characteristics of Original MMDP and Reconstructed Traffic Patterns

	Original	Reconstructed
$E[T_1]$	2.5 sec	2.59 sec
$(\frac{1}{\lambda_1})$	0.00102 sec	0.00102 sec
$E[T_2]$	0.25 sec	0.244 sec
$(\frac{1}{\lambda_2})$	0.01 sec	0.01 sec

Table 3-24 (a). Parameter Values used for Time Comparison Algorithm II

C(T)	T_{nb} (T_{bn})
6	1

Table 3-24 (b). Characteristics of Original On-Off Process and Reconstructed Traffic Patterns

	Original	Reconstructed
$E[T_1]$	2.5 sec	2.64 sec
$(\frac{1}{\lambda_1})$	0.00102 sec	0.001010 sec
$E[T_2]$	0.25 sec	0.257 sec
$(\frac{1}{\lambda_2})$	x	x

The state diagram of a more robust algorithm is shown in Figure 3-19. This algorithm is termed as the "time comparison algorithm III". There are four states: prenormal, normal, prebursty, and bursty. At the start of the algorithm, the current state can be in either prenormal state or prebursty state. Let the initial state be in the prenormal state. At the state, the hypothesis testing is performed. To reduce false transitions and error accumulation, in order for the prenormal state to transit to another state, a certain condition has to be met. In order to transit to the normal state, the number of accumulated successful testing must be greater than N1. However, if the number of accumulated failed testing is greater than N2, the state is transited to the prebursty state. When the testing fails, the new sample is discarded and no state transition is made since the sample is uncertain which state it belongs to. The statistics is collected only when the testing is successful. When the state transits to the normal state, it will stay in the state until there are N3 consecutive failed testing. If this happens, the state transits to the prebursty state. As before, the statistics is collected only when the testing is successful. The same procedure is followed when the state is in the prebursty state. The proposed algorithm can eliminate false transitions and discard the uncertain samples; hence, there is no error accumulation. An experiment using the above algorithm to estimate the original traffic characteristics is performed. The value of C(t) is 6 and the values of N1, N2, N3, B1, B2, and B3 are the same, 2. The reconstructed traffic characteristics for different source models are shown in Tables 3-25 to 3-27.

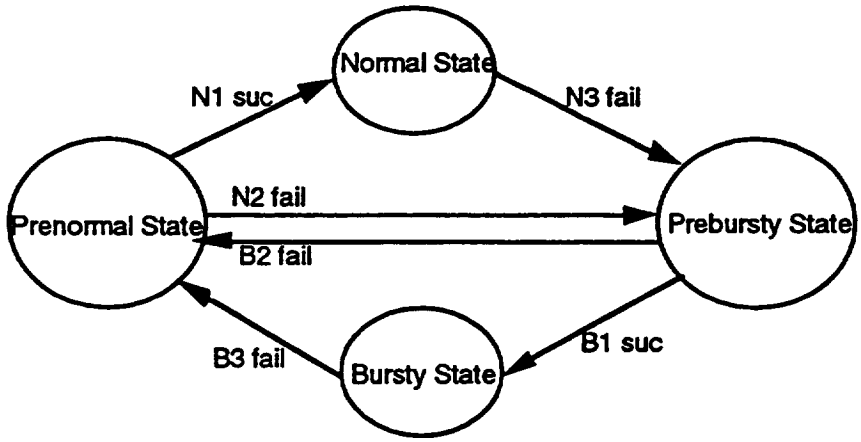

Figure 3-19. State Diagram of Time Comparison Algorithm III

Table 3-25. Characteristics of Original MMPP and Reconstructed Traffic Patterns

	Original	Reconstructed
$E[T_1]$	2.5 sec	2.49 sec
$(\frac{1}{\lambda_1})$	0.00102 sec	0.001029 sec
$E[T_2]$	0.25 sec	0.1848 sec
$(\frac{1}{\lambda_2})$	0.01 sec	0.01122 sec

Table 3-26. Characteristics of Original MMBP and Reconstructed Traffic Patterns

	Original	Reconstructed
$E[T_1]$	2.5 sec	2.76 sec
$(\frac{1}{\lambda_1})$	0.00102 sec	0.001023 sec
$E[T_2]$	0.25 sec	0.2318 sec
$(\frac{1}{\lambda_2})$	0.01 sec	0.0106 sec

Table 3-27. Characteristics of Original MMDP and Reconstructed Traffic Patterns

	Original	Reconstructed
$E[T_1]$	2.5 sec	2.757 sec
$(\frac{1}{\lambda_1})$	0.00102 sec	0.00102 sec
$E[T_2]$	0.25 sec	0.274 sec
$(\frac{1}{\lambda_2})$	0.01 sec	0.01 sec

After the source model is constructed, the accuracy of the model must be checked. For example, after the MMPP is constructed, $r = \frac{1}{E[T_1]} + \frac{1}{E[T_2]}$, $E[\lambda]$, and $E[\lambda^2]$ should be calculated and compared with the measurement results.

Another usefulness of the real time algorithm is that the estimated parameter values can be used as initial values for the offline algorithm. For example, instead of solving Equation 3-35 for r, the estimated r can be use as an initial guess for c in applying the offline fitting algorithm.

Section 4

Traffic Generation Procedure

This section presents packet header contents for a traffic simulator, a traffic flow analysis, a traffic synthesis procedure, and a mechanism of forcing congestion.

4.1 Packet Header Contents

The packet header structure of the mesh VSAT network as given in [1-1] is shown in Figure 2-1. The traffic simulator generates data packets with the basically identical header format as the underlying network. However, some modifications of the header contents may be necessary for the testing purposes. First, the necessary header fields for the traffic simulator are identified and described below. Then, relationships between the proposed header structure and that of the mesh VSAT network are presented.

The packet header structure generated by the traffic simulator is shown in Figure 4-1. The figure also shows the relationships between the packet header fields of the traffic simulator and those of the operational satellite network. A single-satellite scenario is assumed, although its extension to multiple satellites is straightforward. The packet header generated by the traffic simulator consists of the following routing and channel identification information:

- Multicast identification: 1 bit (when this bit is set, the packet is broadcast to all the dwell areas of the selected downlink beams)
- Downlink beam identification: 8 bits (eight independently selected beams for packet routing)
- Dwell area identification: 3 bits (this field is ignored when the multicast bit is set)
- Simulator identification: 3 bits (specifies one of the eight simulators)
- Virtual channel number (VCN): 8 bits (identifies one of 256 unique circuit connections from the selected traffic simulator)
- Packet sequence number (PSN): 4 bits (identifies a specific packet associated with a selected VCN)
- Time stamp: 32 bits (time slot number: $0 \sim 2^{32}$ or $0 \sim 48.5$ days)

Special header fields, which are not available in the original structure, are included in the TBD field. The TBD field may also include FEC parity bits for a packet header.

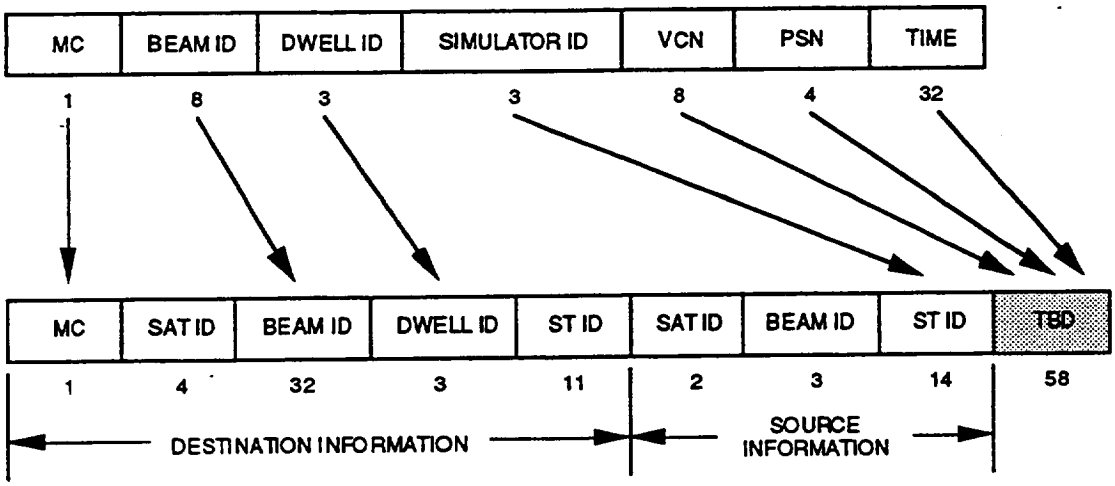

Figure 4-1. Mapping of Packet Header Fields of Traffic Simulator and Satellite Network

If there is no sufficient space available in this field, the time stamp bits may be included in the information field. The traffic simulator generates packet headers of the satellite network with certain fields replaced by the header contents as indicated in the figure. Unassigned fields and unused bits in the assigned fields may contain predetermined bit patterns. Alternately, the traffic simulator may preselect or dynamically select unassigned bits to provided additional features, such as selection of a satellite, identification of source earth terminals, and priority control.

Each traffic simulator can generate up to 256 different circuit connections, and when combined with the simulator IDs a maximum of 2048 distinct channels can be accommodated in the testbed. Each VCN is associated with a specific point-to-point or multicast connection, and multiple VCNs may be assigned to the same beam/dwell area connection.

The packet sequence number is useful for identifying discarded packets by the on-board switch due to congestion. A four-bit field allows detection of up to 15 consecutive packet losses with the same VCN. Although some ambiguity exists in identifying discarded packets if more than 15 packets get lost, this is a very unlikely event.

A time stamp is added to the packet header at the time of packet transmission. Time stamps are useful for measuring the average packet delay and delay jitter through the switch. Measured timing data can be processed in real time or off-line. Interesting measurement, although it complicates traffic simulator design, can be realized by placing a time stamp at the time of packet generation and in emulating earth station buffer operation. This will provide delay performance measurements between user interfaces.

Although not included in the packet header, a priority control field would be a nice feature to be added. Some traffic, such as voice and video, is delay sensitive and requires a low packet loss ratio, and others are less sensitive to the propagation delay and can tolerate a higher packet loss ratio. When congestion occurs the on-board switch discards low priority packets first, providing better performance for high priority

services. This will have an impact on on-board buffer management but enhance user service flexibility.

4.2 Traffic Analysis

Traffic analysis is a process of estimating a traffic flow from traffic simulators to downlink beams for a given set of simulator parameters. Traffic synthesis, discussed in Section 4.3, is a reverse process of deriving traffic simulator parameters from a set of system traffic requirements, such as a system traffic loading factor, downlink beam traffic loading factors, the volume of point-to-point and multicast traffic, and a packet loss ratio. A synthesis procedure uses simple analytical tools to properly select traffic simulator parameters to meet desired system test performance.

4.2.1 Characterization of Traffic Simulator

The number of traffic simulators in the DDPS testbed ranges from one per uplink beam (a total of 8 simulators) to 32 per uplink beam (a total 256 simulators). In the following discussion, the eight-simulator configuration is assumed, since this can be implemented with current technology and simplifies a test setup and simulator operation. It is also desirable to be able to test the DDPS with a smaller number of simulators, for example, in the case of some simulator failure. In an extreme case, a single traffic simulator can be used to fully load all downlink beams using only a 1/8 of its traffic capacity - all packets are broadcast to all dwell areas. Thus, the number of simulators assumed in the analysis is s, where $s \leq 8$. Each simulator is capable of generating a maximum 1024 packets per frame (PPF).

A traffic simulator generates point-to-point (PTP) traffic as well as multicast (MC) traffic. To characterize simulator output traffic, the following notations are used:

C_u Uplink beam capacity (1024 PPF)

C_d Downlink beam capacity (1024 PPF)

T_{ui} Uplink beam average traffic volume (traffic loading) of Simulator i ($T_{ui} \leq C_u$), where $1 \leq i \leq s$ (≤ 8).

p_{ij} Normalized average point-to-point traffic volume from Simulator i to switch output port j, where $1 \leq i \leq s$ and $1 \leq j \leq 8$. Example: $p_{ij} = 0.25$ if a 25% of T_{ui} goes from simulator i to output port j.

t_{ihj} Multicast connectivity value (0 or 1) for Simulator i ($1 \leq i \leq s$), multicast configuration number h ($1 \leq h \leq 247$), and switch output port j ($1 \leq j \leq 8$). (Detailed explanation given below.)

q_{ih} Normalized average traffic volume from Simulator i with multicast configuration h, where $1 \leq i \leq s$ and $1 \leq h \leq 247$. (Detailed explanation given below.)

A traffic simulator may generate multicast traffic with various connectivity configurations, where each configuration represents an uplink-downlink beam interconnection associated with a unique VCN. There are 247 distinct multicast configurations for a given traffic simulator (i.e., 2^8 - 1 no-connection - 8 point-to-point connections). Figure 4-2 shows an example of multicast connections generated by a traffic simulator. In this example, Traffic Simulator i generates four types of beam connectivity.

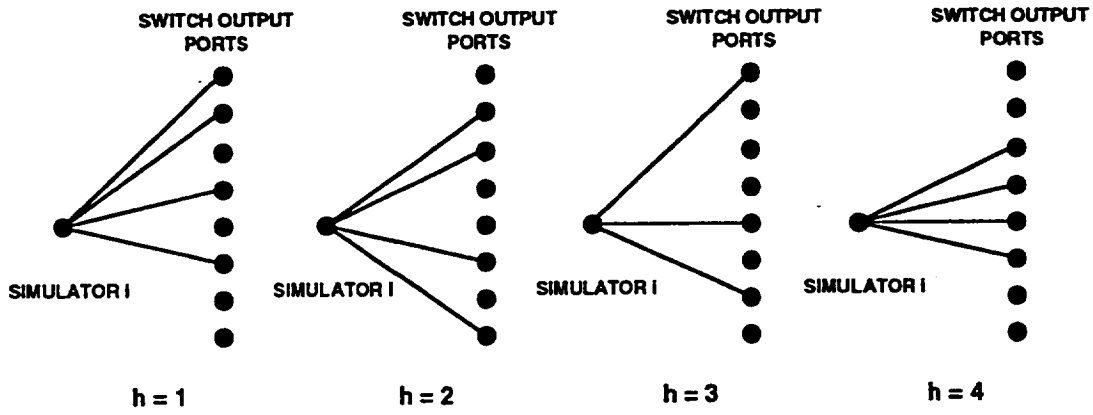

Figure 4-2. Example of Multicast Connections

The connectivity value t_{ihj} for a configuration h is 1 if output port j is a part of the multicast connection; otherwise, it is 0. In the above example, the following connectivity values are assigned:

$$[t_{i11}, t_{i12}, t_{i13}, t_{i14}, t_{i15}, t_{i16}, t_{i17}, t_{i18}] = [1, 1, 0, 1, 0, 1, 0, 0]$$

$$[t_{i21}, t_{i22}, t_{i23}, t_{i24}, t_{i25}, t_{i26}, t_{i27}, t_{i28}] = [0, 1, 1, 0, 0, 1, 0, 1]$$

$$[t_{i31}, t_{i32}, t_{i33}, t_{i34}, t_{i35}, t_{i36}, t_{i37}, t_{i38}] = [1, 0, 0, 0, 1, 0, 1, 0]$$

$$[t_{i41}, t_{i42}, t_{i43}, t_{i44}, t_{i45}, t_{i46}, t_{i47}, t_{i48}] = [0, 0, 1, 1, 1, 1, 0, 0]$$

The normalized average traffic volume q_{ih} is defined for each multicast connectivity configuration h. For example, if the traffic distribution of simulator i is 25% for point-to-point traffic, and 15%, 10%, 20%, and 30% for multicast configurations h=1, 2, 3, and 4, then q_{ih} is given by

$$[q_{i1}, q_{i2}, q_{i3}, q_{i4}] = [0.15, 0.1, 0.2, 0.3]$$

4.2.2 Traffic Flow Equations

Using the traffic simulator parameters defined above, various traffic flow values can be obtained using simple equations. In the following, symbols n, w, m_i, and ρ denote respectively the size of a routing switch (i.e., n = 8), the number of dwell areas per downlink beam (i.e., w = 8), the number of multicast connectivity configurations for

Traffic Simulator i ($m_i \leq 247$), and the packet loss ratio of the switch ($\rho_j < 1$). For multicast traffic, the packets received by a switch output port are broadcast to all dwell areas of the corresponding downlink beam. Thus, a single multicast packet in an uplink results in a maximum of 64 downlink dwell area packets.

Total PTP Traffic to Output Port j: $$P_{oj} = \sum_{i=1}^{s} p_{ij} T_{ui}$$

Total MC Traffic to Output Port j: $$M_{oj} = \sum_{i=1}^{s} T_{ui} \sum_{h=1}^{m_i} q_{ih} t_{ihj}$$

Total Traffic to Output Port j: $$T_{oj} = P_{oj} + M_{oj}$$

Total PTP Traffic to Downlink Beam j: $$P_{dj} = (1 - \rho_j) P_{oj}$$

Total MC Traffic to Downlink Beam j: $$M_{dj} = (1 - \rho_j) w M_{oj}$$

Total Traffic to Downlink Beam j: $$T_{dj} = (P_{dj} + M_{dj})$$

Total Uplink PTP Traffic: $$P_u = \sum_{i=1}^{s} T_{ui} \sum_{j=1}^{n} p_{ij} = \sum_{j=1}^{n} P_{oj}$$

Total Uplink MC Traffic: $$M_u = \sum_{i=1}^{s} T_{ui} \sum_{h=1}^{m_i} q_{ih}$$

Total Uplink Traffic: $$T_u = P_u + M_u = \sum_{i=1}^{s} T_{ui}$$

Total Downlink PTP Traffic: $$P_d = \sum_{j=1}^{n} P_{dj}$$

Total Downlink MC Traffic: $$M_d = \sum_{j=1}^{n} M_{dj}$$

Total Downlink Traffic: $$T_d = P_d + M_d$$

It is often convenient to express a traffic volume in a normalized form. A traffic loading factor is defined as the ratio of an information traffic volume and a total traffic capacity. Several loading factors are defined below:

Traffic Simulator Loading Factor: $$\eta_{ui} = \frac{T_{ui}}{C_u}$$

Total Uplink Loading Factor: $\eta_u = \frac{T_u}{nC_u}$

Downlink Beam Loading Factor: $\eta_{dj} = \frac{T_{dj}}{C_d}$

Total Downlink Loading Factor: $\eta_d = \frac{T_d}{nC_d}$

The downlink beam loading factor η_{dj} is an important indication for predicting a severity of switch output port congestion and provides an estimated packet loss ratio for a given buffer size. One technique to introduce controlled congestion in the the system is to select a desired packet loss ratio and to estimate the corresponding downlink loading factor by a lookup table. This technique is used in the traffic synthesis and explained in Section 4.3.

4.2.3 Example

The example presented herein assumes a DDPS test configuration using two traffic simulators (i.e., s = 2). These simulators generate traffic of the following characteristics:

T_{u1} = 720 PPF, T_{u2} = 960 PPF, $C_u = C_d$ = 1024 PPF

$[p_{1j}]$ = [0.1, 0.15, 0.2, 0.08, 0.22, 0.12, 0.03, 0.02]

$[p_{2j}]$ = [0.07, 0.12, 0.08, 0.15, 0.07, 0.11, 0.24, 0.05]

Multicast connectivity configurations are shown in Figure 4-3. Simulators 1 and 2 generate respectively three and four types of multicast connection, where Simulator 2 includes a broadcast connection. Multicast connectivity values and normalized traffic volumes are given as follows:

$[t_{11j}]$ = [1, 0, 0, 0, 1, 0, 1, 1] $[t_{21j}]$ = [0, 1, 0, 0, 1, 0, 0, 0]

$[t_{12j}]$ = [0, 1, 1, 0, 0, 1, 0, 1] $[t_{22j}]$ = [1, 1, 1, 1, 1, 1, 1, 1]

$[t_{13j}]$ = [1, 1, 0, 1, 0, 0, 1, 0] $[t_{23j}]$ = [0, 0, 1, 1, 1, 0, 1, 0]

$[t_{24j}]$ = [0, 0, 1, 0, 0, 1, 0, 1]

$[q_{1h}]$ = [0.03, 0.04, 0.01] $[q_{2h}]$ = [0.04, 0.02, 0.02, 0.03]

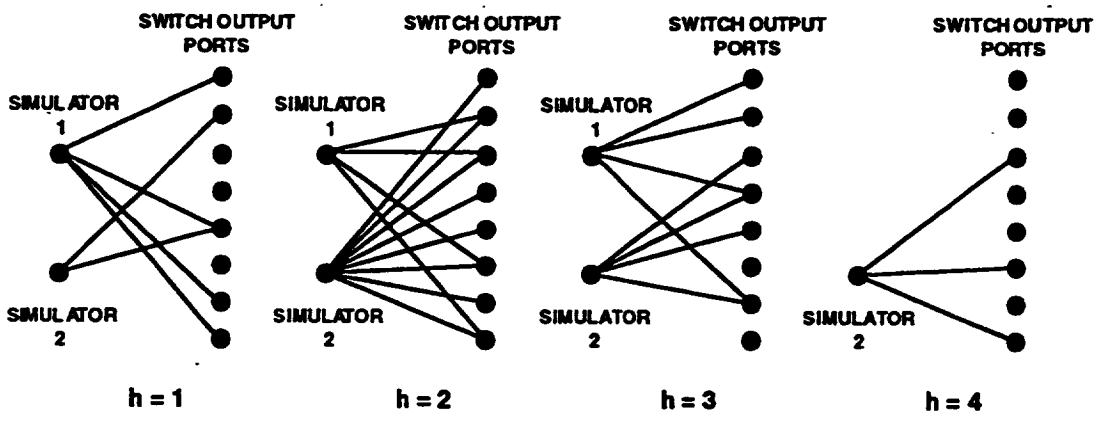

Figure 4-3. Multicast Connections of Traffic Simulators 1 and 2

The results of the analysis are summarized in Table 4-1. For simplicity, no packet loss ($\rho = 0$) is assumed in the analysis. A few interesting facts should be pointed out. The total uplink multicast traffic is 163 PPF which is about a 10-percent of the total uplink traffic volume. These packets are multicast to different output ports and further broadcast to all the dwell areas associated with the output ports. Thus, the contribution of uplink multicast traffic to the total downlink traffic becomes significantly larger,

Table 4-1. Summary of Traffic Parameters for the Example

$[P_{oj}]$	139	223	221	202	226	192	252	62
$[M_{oj}]$	48	94	96	46	98	77	67	98
$[T_{oj}]$	187	317	317	247	324	269	319	161
$[P_{dj}]$	139	223	221	202	226	192	252	62
$[M_{dj}]$	384	749	768	365	787	614	538	787
$[T_{dj}]$	523	972	989	566	1013	806	790	850
P_u	1517							
M_u	163							
T_u	1680							
P_d	1517							
M_d	4992							
T_d	6509							
$[\eta_{ui}]$	0.70	0.94						
η_u	0.21							
$[\eta_{dj}]$	0.51	0.95	0.97	0.55	0.99	0.79	0.77	0.83
η_d	0.79							

i.e., 4992 PPF which is a 77-percent of the total downlink traffic. In the operational system, the amount and connectivity of multicast traffic must be strictly controlled to avoid switch congestion. The table shows a traffic loading factor of 0.99 in downlink beam #5. However, this value may not be achievable in the actual test, in particular to meet a low packet loss ratio.

4.2.4 Special Case Analysis

To derive explicit expressions for various traffic flow parameters, a special case is considered in this subsection. The results presented below are useful for an understanding of the problem. Consider s traffic simulators ($1 \le s \le 8$), each generating traffic of the same characteristics. All point-to-point traffic generated by a simulator are equally distributed to all output ports, and all multicast traffic are broadcast to all downlink beams. Let r be the traffic loading factor of a simulator, and p and q be the total normalized average traffic volumes of point-to-point and multicast traffic, respectively. Then, the following relationships hold:

$$p + q = 1$$

$$\eta_{ui} = r,\ 1 \le i \le s$$

$$T_{ui} = rC,\ 1 \le i \le s \text{ (note: } C_u = C_d = C)$$

$$p_{ij} = p/n,\ 1 \le i \le s \text{ and } 1 \le j \le n$$

$$m_i = 1,\ 1 \le i \le s$$

$$t_{ihj} = 1 \text{ for } h = 1 \text{ and } 1 \le i \le s \text{ and } 1 \le j \le n;\ = 0, \text{ otherwise}$$

$$q_{ih} = q \text{ for } h = 1 \text{ and } 1 \le i \le s;\ = 0, \text{ otherwise}$$

Table 4-2 summarizes the results of the analysis. The table also includes parameter values for two extreme cases: point-to-point only traffic (i.e., $p = 1$ and $q = 0$) and multicast-only traffic (i.e., $p = 0$ and $q = 1$).

The total downlink traffic is given by $(1 - \rho)(p + nwq)srC$ for mixed point-to-point and multicast traffic. The sensitivity of multicast traffic to the total downlink traffic is $nw = 64$ times greater than that of point-to-point traffic. A similar effect was also observed in the previous example.

Table 4-2. Summary of Traffic Parameters for a Special Case

TRAFFIC FLOW	SYMBOL	PTP & MC	PTP (p=1, q=0)	MC (p=0, q=1)
Total PTP Traffic to Output Port j	P_{oj}	sprC/n	srC/n	
Total MC Traffic to Output Port j	M_{oj}	sqrC		srC
Total Traffic to Output Port j	T_{oj}	(p/n + q)srC	srC/n	srC
Total PTP Traffic to Downlink Beam j	P_{dj}	(1 - ρ)sprC/n	(1 - ρ)srC/n	
Total MC Traffic to Downlink Beam j	M_{dj}	(1 - ρ)swqrC		(1 - ρ)swrC
Total Traffic to Downlink Beam j	T_{dj}	(1 - ρ)(p/n + wq)srC	(1 - ρ)srC/n	(1 - ρ)swrC
Total Uplink PTP Traffic	P_u	sprC	srC	
Total Uplink MC Traffic	M_u	sqrC		srC
Total Uplink Traffic	T_u	srC	srC	srC
Total Downlink PTP Traffic	P_d	(1 - ρ)sprC	(1 - ρ)srC	
Total Downlink MC Traffic	M_d	(1 - ρ)nswqrC		(1 - ρ)nswrC
Total Downlink Traffic	T_d	(1 - ρ)(p + nwq)srC	(1 - ρ)srC	(1 - ρ)nswrC
Traffic Simulator Loading Factor	η_{ui}	r	r	r
Total Uplink Loading Factor	η_u	r	r	r
Downlink Beam Loading Factor	η_{dj}	(1 - ρ)(p/n + wq)sr	(1 - ρ)sr/n	(1 - ρ)wsr
Total Downlink Loading Factor	η_d	(1 - ρ)(p/n + wq)sr	(1 - ρ)sr/n	(1 - ρ)wsr

4.3 Traffic Synthesis

Traffic synthesis is the reverse process of traffic analysis and may be used along with the analysis procedure to properly select traffic simulator loading parameters. The use of the analysis technique alone complicates a parameter selection process for achieving a desired system performance level for testing. The synthesis procedure described in the following should be taken as a representative, since there is no unique way of selecting a set of simulator parameters to achieve the desired performance goal.

4.3.1 Downlink Traffic Loading and Packet Loss Ratio

In the DDPS system, one of the critical operational considerations is a congestion control mechanism to prevent performance degradation due to switch congestion. A certain packet loss ratio (PLR) must be met for a given service. For example, a circuit switched service should have a very low PLR, desirably no packet losses due to congestion, while some packet switched traffic may tolerate a larger PLR. The PLR is directly related to the downlink traffic loading factor (DLLF) for a given size of on-board buffer. Figure 4-4 plots the relationship between the DLLF and PLR for various buffer sizes. Poisson traffic distribution is assumed. A similar curve may be obtained for other

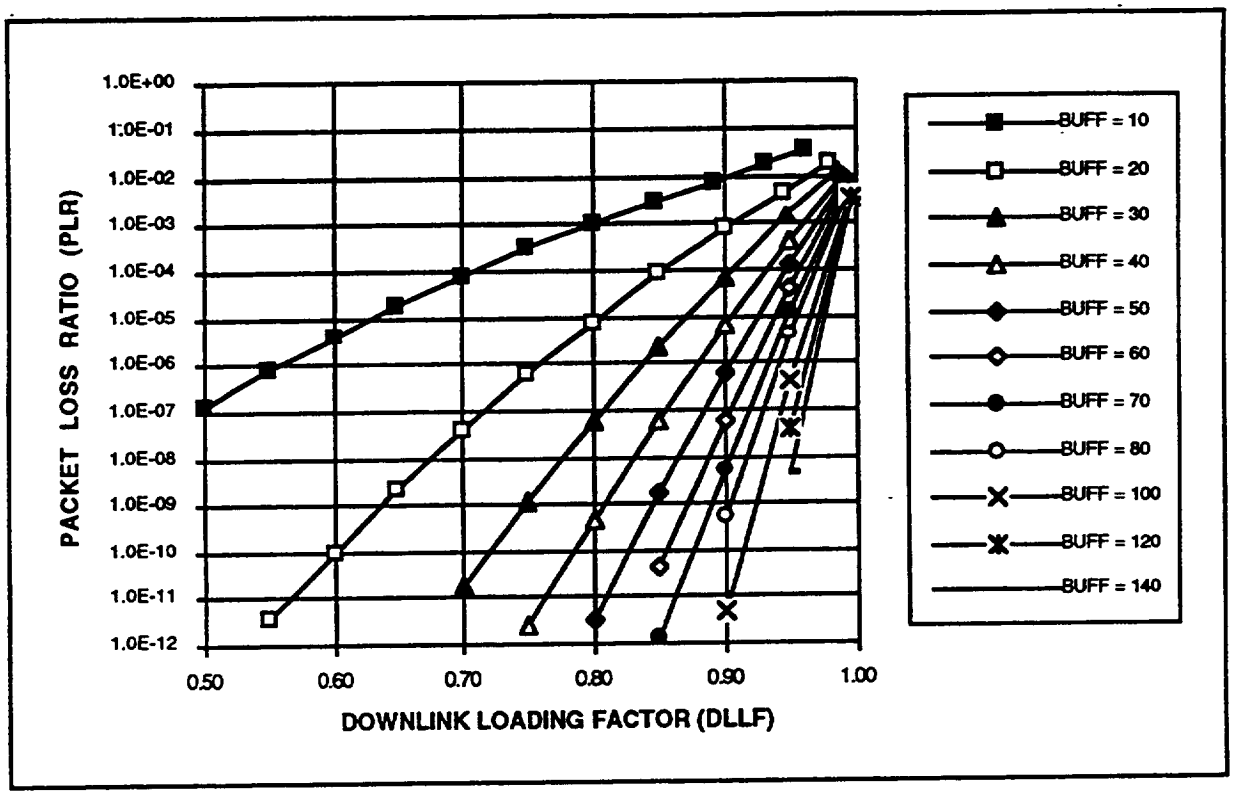

Figure 4-4. Relationship Between DLLF and PLR for Buffer Size as Parameter

types of traffic, such as bursty traffic [4-1]. The traffic synthesis procedure described below uses this relationship to select proper simulator parameters for a given degree of congestion.

The DLLF-PLR curve given in the figure may also be applicable to dwell area traffic depending on the buffer structure and the time slot allocation procedure used for downlink transmission. Six cases are shown in Table 4-3. A common buffer is shared by all dwell area traffic and has no predefined individual buffer allocation to different dwell areas. Separate dwell area buffers may consist of a single physical storage area

Table 4-3. Alternate Dwell Area Traffic Buffer Designs

Buffer Structure	Buffer Allocation	Downlink Time Slot Allocation	
		Dynamic	Preallocation
Common Downlink Buffer	Dynamic	Case 1	Case 2
	Preallocation	N/A	N/A
Separate Dwell Area Buffers	Dynamic	Case 3	Case 4
	Preallocation	Case 5	Case 6

for a downlink beam, but its partition to different dwell area traffic can be allocated dynamically or on a preallocation basis. In preallocation, the available resources are adjusted at a much slower rate than the frame period to accommodate changes in the overall traffic flow. A time slot allocation procedure can also be dynamic based on the amounts of dwell area traffic or can be preallocated. Different DLLF-PLR performance will result in depending on the allocation procedure used.

Cases 1 and 3 utilize the allocated buffers most efficiently and yield the same DLLF-PLR performance, provided that dwell area buffer allocation and time slot allocation in Case 3 are properly coordinated. The DLLF-PLR curve shown in Figure 4-4 can used to estimate the PLR for downlink beam and dwell area traffic for a given DLLF and a buffer size.

In Case 2, the DLLF-PLR performance will vary depending on the amounts of traffic to different dwell areas and their dwell durations. An analysis or simulation is needed to characterize the relationship among the PLR, dwell area traffic loading, and dwell area capacity.

Case 4 is a more general case of Case 2 in which dynamic buffer allocation is performed according to a certain rule based on dwell area traffic volume. As a special case, in which dwell area buffer sizes are adjusted on a packet-by-packet basis for incoming traffic, the PLR performance becomes identical to that of Case 2.

In Cases 5 and 6, dwell area traffic buffers are preallocated, and downlink time slots are either dynamically allocated or preallocated. For preallocated time slot operation (Case 6), the relationship given in Figure 4-4 can be used to estimate the PLR for a given buffer size, where the DLLF should be interpreted as a dwell area traffic loading factor. The PLR performance for Case 5 will be better than that of Case 6 and will require a further analysis to take into account a specific time slot allocation procedure.

Depending on the design technique used for downlink dwell buffers, the proper DLLF-PLR performance curve should be used in the synthesis procedure. In the following, Case 1 or Case 2 is assumed for simplicity. Consideration should also be given to the impact of the additional buffer required to absorb the time displacement caused by downlink time plan changes.

4.3.2 General Procedure

A general procedure for traffic synthesis is depicted in a flow diagram in Figure 4-5. The procedure consists of three major routines: (a) uniform traffic loading, (b) simulator traffic adjustment, and (c) DLLF/PLR adjustment. Uniform traffic loading generates a set of traffic simulator parameters which meets a specified DLLF/PLR value and is

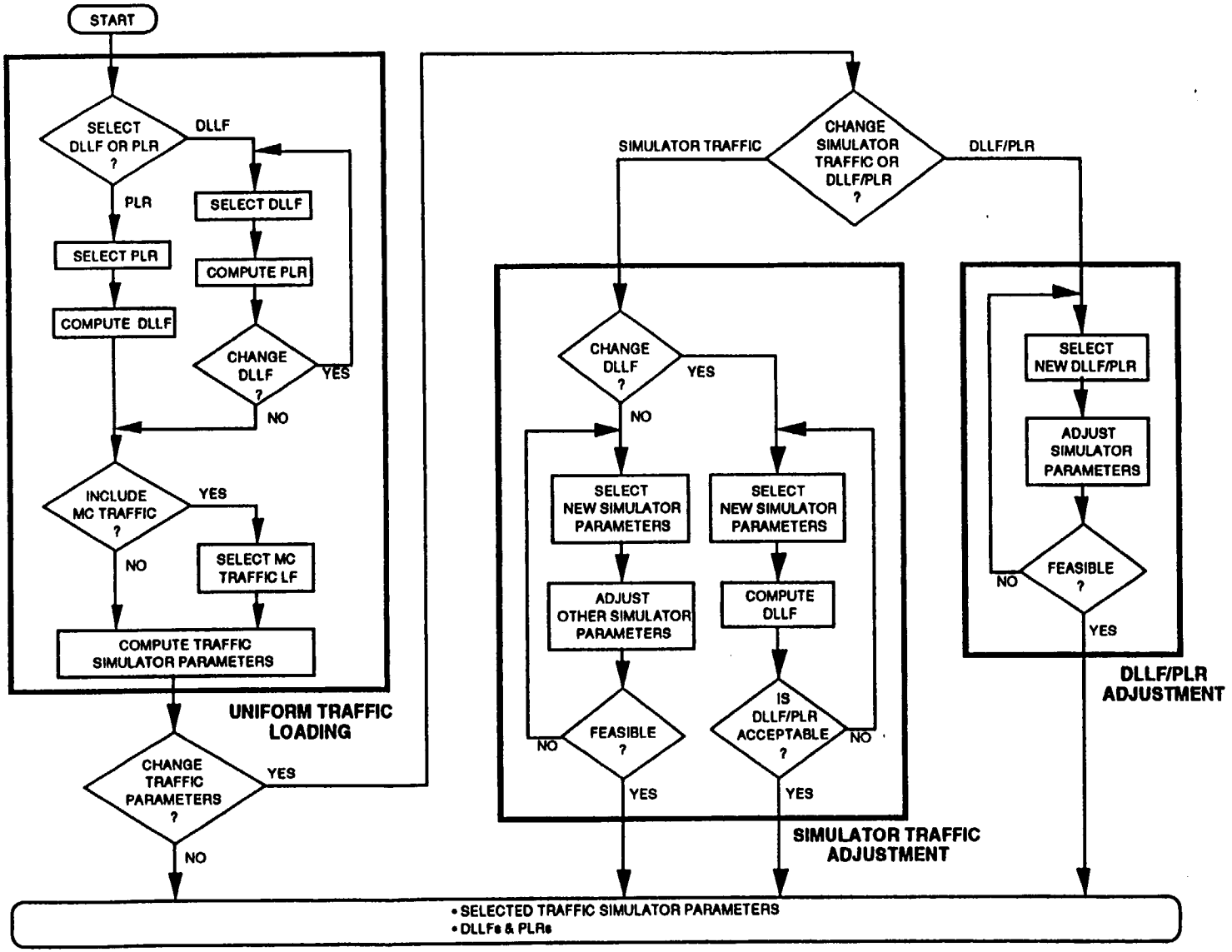

Figure 4-5. Traffic Synthesis Flow Diagram

based on uniform traffic distribution of PTP and broadcast traffic. The resulting traffic profile may be modified to generate various other traffic scenarios. The simulator traffic adjustment step allows modifications of traffic simulator parameters with/without changes in the traffic loading factors of downlink beams. The DLLF/PLR adjustment routine, on the other hand, provides a capability of selecting a different DLLF/PLR for each downlink beam. The major difference in the computational procedure between the two routines is that the former is primarily based on foreword calculation (analysis) and the latter backward calculation (synthesis). Detailed computational procedures of these routines are described in the following. Although the proposed synthesis procedure should satisfy most of the testbed operational requirements, other features may be accommodated as needed.

Uniform Traffic Loading

Uniform traffic loading is a procedure of setting up traffic simulator parameters to meet a desired DLLF or PLR. All traffic simulators generate uniformly distributed traffic to all the downlink beams and broadcast to all dwell areas for multicast traffic. Let η_d be the DLLF corresponding to a desired PLR. Furthermore, assume that α and β are the normalized distributions of point-to-point and multicast traffic in the downlink beam: $\alpha + \beta = 1$, and $\alpha\eta_d C$ and $\beta\eta_d C$ are respectively the average downlink beam PTP and MC traffic volumes. The downlink traffic is equally distributed to all dwell areas within a beam.

Simulator parameters to satisfy the above traffic distribution are obtained from Table 4-2. In the following equations, various notations are defined in Subsection 4.2.4, and eight downlink beams and eight dwell areas per beam are assumed ($n = w = 8$).

Simulator Traffic Loading Factor:
$$r = \frac{[1 + \alpha(nw - 1)]\eta_d}{(1 - \rho)ws} = \frac{(1 + 63\alpha)\eta_d}{8(1 - \rho)s}$$

Normalized Average PTP Traffic Volume:
$$p = \frac{\alpha wn}{(1 - \rho)[1 + \alpha(wn - 1)]} = \frac{64\alpha}{(1 - \rho)(1 + 63\alpha)}$$

Normalized Average MC Traffic Volume:
$$q = 1 - p = \frac{1 - \alpha}{(1 - \rho)[1 + \alpha(wn - 1)]} = \frac{1 - \alpha}{(1 - \rho)(1 + 63\alpha)}$$

For a given number of traffic simulators, the selection of the loading factor (η_d) and normalized PTP traffic volume (α) cannot be arbitrary, since the amount of traffic generated by a traffic simulator is limited, i.e., $r \leq 1$. Figure 4-6 plots the minimum number of traffic simulators needed in order to test various combination of a downlink loading factor and PTP traffic volume. No packet loss due to congestion is assumed ($\rho = 0$). Thus, the DLLF values shown in the figure can be interpreted as the traffic loading factor at the input to a TDM buffer which closely approximates the DLLF for a small ρ.

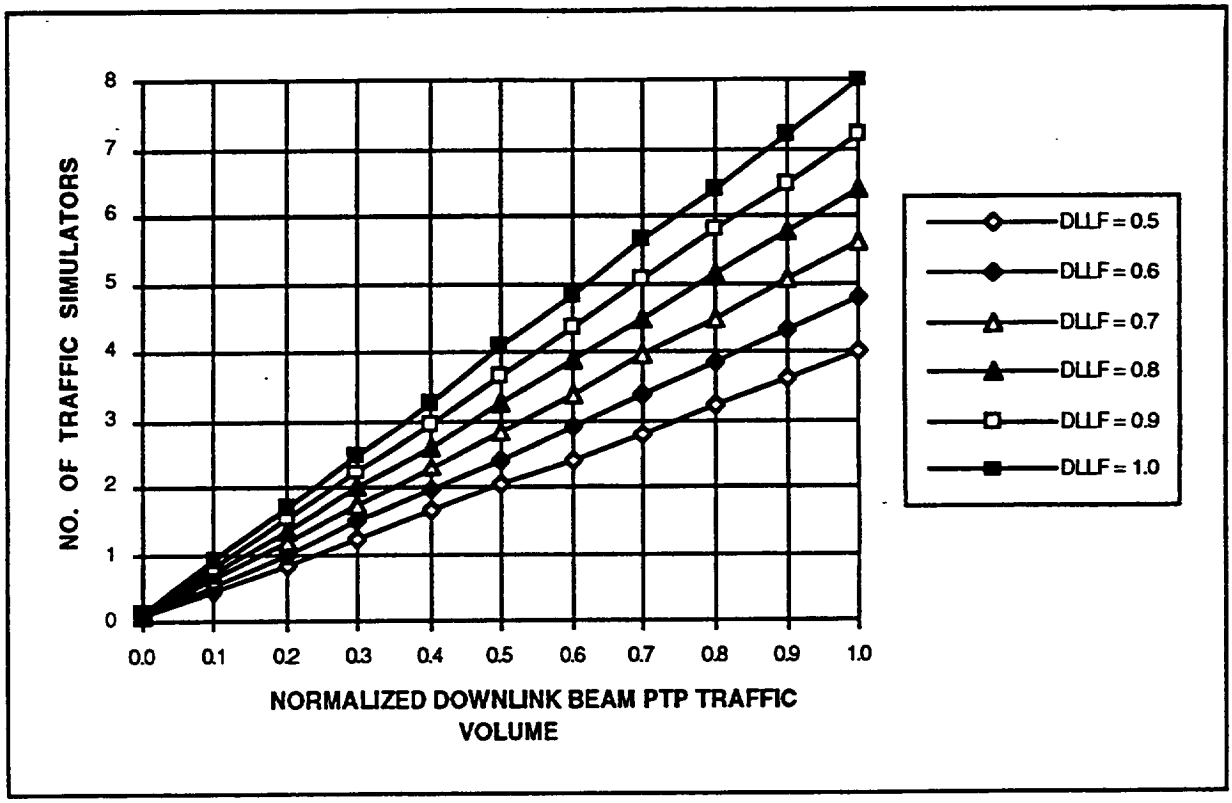

Figure 4-6. Number of Traffic Simulators Required for Given DLLF and PTP Traffic

From the figure, for example, three traffic simulators can generate close to a 100% of downlink traffic of which about a 37% is point-to-point traffic and the rest broadcast traffic.

The traffic simulator parameters computed above generate the required downlink traffic to meet a selected loading factor. These parameters may be modified to test the DDPS system under various other traffic conditions. The following two routines provide guidelines for implementing additional features in the testbed. Procedures for control and operation of traffic simulators and traffic configurations may incorporate the three basic routines for flexible DDPS testing.

Simulator Traffic Adjustment

This step allows the operator to modify the traffic simulator parameters obtained in the above step to test the system for a specific user traffic environment. The operator may wish to modify some of the simulator parameters with or without changing downlink loading factors. In general, changes in simulator parameters will result in different loading factors for different downlink beams. However, an iterative routine based on the traffic analysis can be developed such that changes in some traffic parameters automatically adjust parameters of other traffic simulators to maintain the same downlink loading factors. As in the previous case, a feasibility of a new set of parameters must be examined.

Loading Factor/PLR Adjustment

This routine differs from the simulator traffic adjustment routine in the selection of traffic parameters. The operator selects a new set of downlink loading factors (or packet loss ratios) for individual downlink beams. A built-in computational routine automatically adjusts simulator parameters to yield the selected loading factors. Unrealizable requests will be prompted to the operator with suggestions for possible modifications. The operator does not have a direct access of modifying simulator parameters.

4.3.3 Forcing Multicasting/Broadcasting

A mechanism for forcing multicasting and broadcasting is built into the header structure of data packets. When a multicast bit is set, the packet will be routed to the proper switch output ports according to the beam ID field. These packets are broadcast to all dwell areas of the designated downlink beams. The impact of multicast/broadcast traffic on the downlink beams is evaluated in Section 4.2.2 and analyzed in detail in Sections 4.2.3 and 4.2.4. A traffic synthesis procedure for an arbitrary set of DLLFs with multicast traffic is slightly more complex. Traffic simulator parameters to satisfy the given DLLFs are obtained by solving a set of linear equations given in Section 4.2.2. Three possible cases exist:

a. There is no feasible set of parameters.

b. A unique solution exits.

c. There are more than one set of parameters to satisfy the given DLLFs.

In most cases, the solution will be either a or c. If there is no feasible solution, a new set of DLLFs should be selected. If there is no unique solution, some simulator parameters may be arbitrarily selected within certain constraints such that the selected parameters produce the desired downlink loading factors.

4.3.4 Inducing Beam/Dwell Congestion

A mechanism for inducing congestion at the selected downlink beam or dwell area is also easily accommodated by increasing the amount of simulator traffic to the selected downlink beam or dwell area. A detailed discussion is provided in the previous sections. For random traffic, potential congestion always exists regardless the size of downlink buffer and the amount of traffic loading. Thus, inducing congestion may be interpreted as increasing the severerity of congestion, or equivalently increasing a packet loss ratio. Figure 4-4 provides a guideline for selecting the DLLF for a given PLR, and the traffic synthesis procedure described in the previous section generates a set of traffic simulator parameters to cause the required congestion state. As in the case of multicast traffic, some traffic simulator parameters must be properly selected in solving linear equations.

4.4 Congestion Control

A congestion control procedure consists of four steps: a) monitoring, b) processing, c) transmission of control messages, and d) implementation. Congestion monitoring is performed by the DDPS, and the status information is sent to the network control system. The network control system processes the congestion status information to determine the timing for exercising congestion control. It also generates proper control messages for the traffic simulators. The traffic simulator responds to the congestion control messages by increasing or decreasing traffic to certain downlink beams and dwell areas. In the operational system, the last three steps may be implemented by the user terminals. Alternately, the on-board controller may implement the second step for centralized control. The following sections describe the first three steps in detail, and the fourth step is discussed in Section 5.

4.4.1 Congestion Monitoring

The state of switch congestion may be monitored using one of two techniques. The first technique is to directly measure the length of the packet queue in the TDM buffer. The queue length information is periodically, for example once a frame, sent to the network control system for processing. The change in the queue length between two successive measurements corresponds to the difference in the numbers of transmitted packets and arriving packets during the measurement interval. Since the rate of downlink transmission is known, the rate of packet arrival is easily estimated from the measured data.

The second technique is to count the number of packets arriving at the buffer for a measurement period. The current queue length can be estimated from the previous one by adjusting it for the difference in the numbers of packet receptions and transmissions. Thus, the two techniques basically provide the same congestion information. However, the second technique is less robust because of the following reasons. When congestion occurs and packets are discarded, the number of packets discarded must be taken into consideration in estimating the queue length. Also, if a congestion monitor message gets lost due to transmission errors, the ground terminal loses an accurate count of queued packets.

Dwell area congestion can be monitored using the same techniques given above. Depending on the buffer structure used, either beam congestion monitoring or dwell area congestion monitoring is employed.

4.4.2 Congestion Processing

The network control system receives measurement data, makes a decision on congestion control based on its internal algorithm, and supplies control information to the traffic simulators. These are basically software functions and have flexibility of experimenting different procedures. Since detailed congestion control procedures and simulation results have been presented in the previous study report [4-1], this section describes an overall procedure and considerations of implementing congestion control algorithms.

Queue lengths measured on board may be directly compared against preselected congestion control threshold values (L_1 and L_2, where $L_1 < L_2$) on a frame-by-frame basis. When the queue length of the downlink buffer j exceeds L_2, congestion is declared for the downlink beam (or dwell area), and simulator traffic to that beam is reduced to αT_{ij}, where $\alpha (< 1)$ is a preselected adjustment factor for traffic reduction and T_{ij} is the current traffic volume from Simulator i to downlink beam j. The congestion "on" state is maintained till the queue length decreases below L_1. When this occurs, the congestion state changes to "off", and simulator traffic is increased to βT_{ij}, where $\beta (> 1)$ is a preselected adjustment factor for traffic increase. The use of two threshold values minimizes a "flip-flop" effect of congestion control due to fluctuation of measured queue lengths.

The basic congestion control concept described above may be enhanced with the incorporation of more elaborate procedures for congestion detection and control. For example, rather than using instantaneous queue length values, measured data may be smoothed with a fixed window or sliding window technique. Also, some mechanism to predict the trend of traffic growth or decrease may be useful to optimize the control loop performance (i.e., minimizing congestion while maximizing the throughput). Congestion control is typically performed no more than once per one round trip time based on the observation of the effect of the previous control message. However, for a predictable traffic change, execution of several correction steps in one round trip time will provide better performance. Alternately, more than two threshold values may be established for finer control. Effectiveness of these procedures may be evaluated in the experiment.

4.4.3 Control Message Transmission

To develop a flexible congestion control configuration, most processing tasks are performed by the network control system. The interface between the network control system and traffic simulators is relatively simple and carries the following control messages from the network control system to the traffic simulators:

a. Identification of the downlink beam or dwell area for congestion control, and

b. Traffic volume adjustment factor, α or β.

The traffic simulators simply implement the control message by decreasing or increasing traffic to the designated beam/dwell area by the amount indicated by the adjustment factor. To emulate the operational satellite system environment, the network control system delays the transmission of control message by one round trip time.

Section 5

Traffic Generator Implementation

This section presents block diagrams of high-level design and implementation for a traffic generator (TG). The software/hardware requirements are investigated. Also discussed is the congestion detection procedure and the monitoring functions at the switch as well as the necessary functions of the TG to provide traffic source response procedure to (beam) congestion. A survey of the available traffic generators and their applicability to ISP testing is presented. The overall testbed configuration is presented along with the interactions among different components in the testbed.

There are two basic functions of a traffic generator:

1. packet arrival time generation.

2. packet content (header and payload) generation.

The packet header contains source address, destination address(es), priority, control, FEC, and the testing field. The testing field contains a VCN, SN, and time stamp. The testing field is used for performance measurement. If the packet header size is not large enough to include the testing field, the testing field may be included in the packet payload. The following discussion assumes that the testing field is included in the packet header.

Different approaches of implementing the above two functions in a traffic generator are considered as follows. Depending on the objectives of the testing, three different approaches can be used for traffic generation. First, a commercially available traffic generator or bit pattern generator is used. The major disadvantage is that these generators are not very flexible. For the packet arrival time generation, either the packet arrival time is a constant (for example, packets are read from the RAM following a specific sequence) or only a very few statistical queueing models are available. For the packet header generation, the number of different packet headers is limited and/or it is very specific to certain traffic types (for example ATM). The second approach is to build a static traffic generator in house. The static traffic generator reads packets from a RAM following a deterministic sequence repeatedly. The packet arrival time is deterministic. The number of packet headers which can be specified is limited by the RAM size. Although the static generator can not evaluate the switch performance (such as loss and delay) and the effectiveness of the congestion control algorithm, it can be used to verify the switch operation (such as switch routing). The third approach is to build a dynamic traffic generator. This section will focus on the hardware/software implementations of a dynamic traffic generator. For real time operation, the implementation of the TG should be as hardware oriented as possible.

Before different implementations of a dynamic traffic generator are presented, two important issues are discussed: the uplink rate and the number of traffic generators. The number of traffic generators required largely depends on the role of the traffic generator in the satellite network. If one traffic generator is equivalent to one terminal, then the number of traffic generators is 8 x 1024, which is too excessive. If one traffic generator represents one carrier, then the number of traffic generator is 8 x 32. In this case, each TG is shared by at most 32 terminals since there are only thirty-two 64-kbit/sec slot in one carrier (assuming that there is no subchannel allocation). As discussed in Section 3, eight hardware multiplexer devices need to be built in front of the switch to multiplex uplink carriers into a single high-speed TDM stream. This may increase the cost of the switch and make the analysis of the traffic pattern more difficult. The reason is that even though the traffic pattern for each carrier is known in advance, the statistical multiplexed traffic pattern is not known.

If one traffic generator generates the aggregate traffic from all the terminals in one uplink beam, there are only eight traffic generators. The output of the traffic generator can be directly interfaced with the input port of the switch. The correspondence between the role of the traffic generator and the number of traffic generators is listed in Table 5-1.

Table 5-1. Correspondence between Role of Traffic Generator and Number of Traffic Generators

Role	Number
terminal	8192
carrier (2 Mbit/sec)	256
beam	8

As indicated in Reference 1-1, the uplink rate in the operational system is 65.536 Mbit/sec. Since the size of the packet is 2048 bits, each TG has to be able to generate packets at 32 kpkt/sec. This rate determines whether it is feasible for the TG to generate the traffic in real time. This rate along with the testing duration and the size of the packet header determines the amount of storage required if the traffic is generated off-line. Most commercially available CPU board can only generate packets with a rate in the range of 10 Mbit/sec. This constraint is due to different factors such as the CPU speed and the number of CPU cycles required to generate packet arrival times following a certain distribution. In order to generate packets with the desired speed, special CPU such as a RISC processor or special hardware must be used. If the traffic profile of the traffic generator is constructed off-line, then the speed of generating a packet arrival time is not a concern. To generate 65.536 Mbit/sec real time data, the RAM access cycle is only 2.048 MHz using a 32 bit parallel bus. The parallel to serial converter (between the RAM and the switch) is widely available in the 65.536 MHz range.

5.1 On-Line Dynamic Traffic Generation

The on-line dynamic traffic generation requirement is listed in Table 5-2.

Table 5-2. On-Line Traffic Generation Requirement

Requirement	Value
speed (bit/sec)	65.536 Mbit/sec
speed (pkt/sec)	32 kpkt/sec

There are two approaches, A and B, for on-line dynamic traffic generation. Approach A is to use a workstation (or a PC) to generate real-time traffic. The possible configuration of the TG is shown in Figure 5-1. The workstation must have a physical connection to the outside world. The workstation has an user interface. The user interface should be flexible enough such that the user can set up the test configuration, select the source queueing model for each traffic simulator, characterize the parameter values for each queueing model, enter the duration for the testing, and select the performance measurements. If the workstation is able to handle the generation of individual packets, it generates arrivals of fixed-size packets by executing some traffic source queueing algorithms and assigning the fields of a packet header. Some of the fields in the packet header such as the destination and the priority can be generated by applying different probabilities. If the workstation can not handle each individual packet, the workstation may generate arrivals of larger packets, for example an image or a frame. The workstation generates arrivals of large, variable-length packets by executing simple traffic source queueing algorithms and their associated packet headers. The host-network interface segments the variable-length packets into fixed-length packets and attaches packet headers. The interface also performs FEC and idle cell insertion to maintain link synchronization. The host-network interface chips (most of them are for ATM applications) are available in the market and the speed can go up to 155.52 Mbit/sec. To minimize the cost, a workstation may have to generate traffic for more than one uplink beam. The on-line approach is more feasible if the speed requirement is low. The hardware/software requirements are listed in Table 5-3.

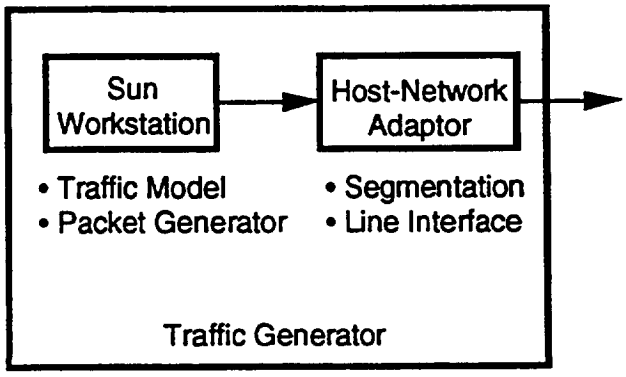

Figure 5-1. Traffic Generator Configuration A

Table 5-3. Hardware/Software Requirement for Configuration A

Hardware	Software
workstation to generate packet arrivals and packet headers on-line	(simple) statistical queueing model
high-speed bus	interface driver
host-network adaptor to segment variable-size packets into fixed-size packets, to attach the packet header, to perform FEC on packet header, and to insert idle packets	

Approach B is to use the workstation to load the burst traffic profile of some statistical queueing arrival process to a processor board off line (see Figure 5-2). The processor board generates packet arrival time based on the burst traffic profile using hardware mechanisms. An example is given below.

A 27 Mbit/sec on-line ATM TG was proposed in Reference 5-1. The TG generates on-off process traffic on-line and in real time. An on-off process is characterized by the on state duration, off state duration and on state arrival rate. The on state packet arrival rate is a constant. The on- and off-state duration is a function of time. The burst traffic profile of each on-off process is constructed off-line by the workstation. The burst traffic profile format is shown in Figure 5-3. The traffic profile is a series of bursts and each burst consists of two elements: the number of information packets and the number of idle packets. The TG generates real-time traffic using these two parameters. At the start, the TG generates burst 0. The arrival time of information packets is generated using the peak rate. If the peak packet arrival rate is equal to the link speed/packet size, there is no gap between two information packets.

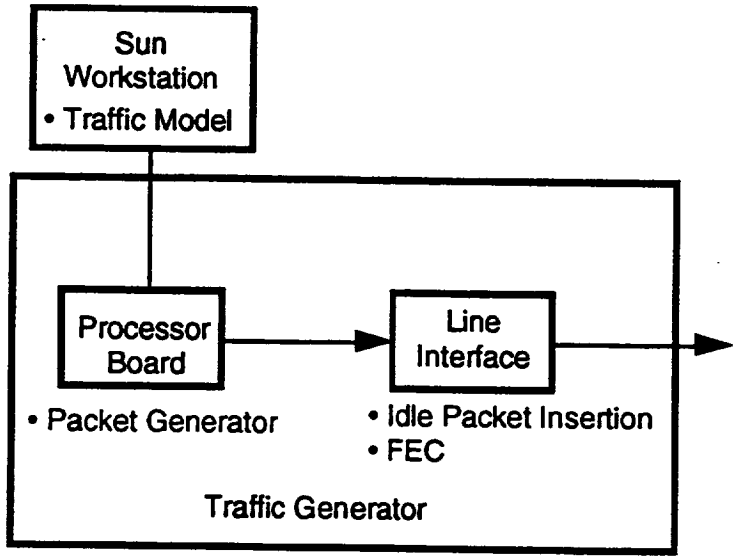

Figure 5-2. Traffic Generator Configuration B

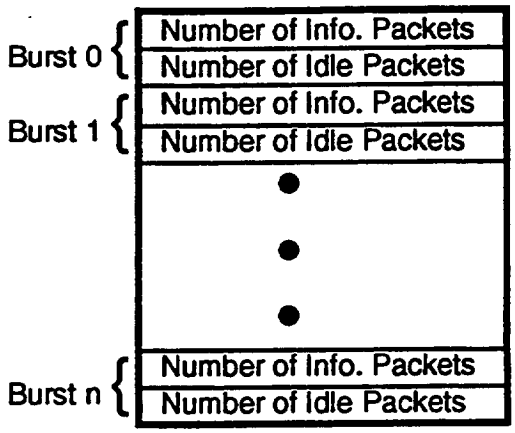

Figure 5-3. Burst Traffic Profile Format for On-Off Process

A counter is implemented to count the number of information packets generated. When the counter value reaches the specified number of information packets in the table, generation of information packets stops. Then the TG generates the idle packets. The same implementation is used to count the number of idle packets. When the counter value reaches the number of idle packets specified in the table, generation of idle packets stops. Then the TG fetches the parameters for burst 1. The procedure repeats itself.

This procedure can be directly applied to this study. Let the peak rate for the on-off process be 32 kpkt/sec. The packet slot time is 31.25 μs. Within 31.25 μs, the processor must generate the packet header, which includes source address, destination address(es), priority, control, FEC and other testing fields.

There are two methods to generate the packet header of an information packet. The first method is to generate the packet header on-line. The processor must be able to generate source address, destination address(es), priority, control, FEC and the testing field. The second method is to store (a small amount of) sample packets in a file. The processor simply fetches the packet in the file either randomly, following a specific sequence, or following a certain distribution. The techniques of generating a random number or generating a sample from a given distribution will be introduced later. The header fields which can be pregenerated are source address, destination address(es), priority, control and VCN. For example, if there are only point-to-point packets and the destination distribution is uniform, the file can contain sixty-four packets, one packet is for one dwell in one downlink beam. One packet is associated with a distinct VCN. Every time a packet header needs to be generated, the processor picks one of the sample packets randomly. However, since some of the test fields (for example the sequence number and the departure time stamp) can not be determined until generation of a packet really occurs, the SN and the departure time stamp have to be inserted into the testing field in the packet header on-line and in real time.

The same concept can be extended to generate the MMDP packet arrivals. An MMDP packet arrival process is characterized by four parameters: state 1 duration, state 1 arrival rate, state 2 duration, and state 2 arrival rate. The burst traffic profile format is shown in Figure 5-4. The state duration is a function of time. The burst traffic profile is

a series of bursts and each burst consists of two elements: the number of information packets for state 1 and the number of information packets for state 2. At the start, the TG generates burst 0. The TG uses the state 1 (constant) arrival rate to generate information packets. When the number of information packets generated reaches the specified number in the table, generation of information packets for state 1 stops. The TG starts generating information packets for state 2. When the number of information packets reaches the specified number in the table, generation of information packets for state 2 stops. Then the TG fetches the parameters for burst 1. The procedure repeats itself. Since the packet arrival rate may be smaller than the link transmission rate, there are gaps between the information packets. A line interface is required to synchronize the uplink transmission time of information packets, fill the gaps between two information packets with idle packets and perform FEC on the packet header.

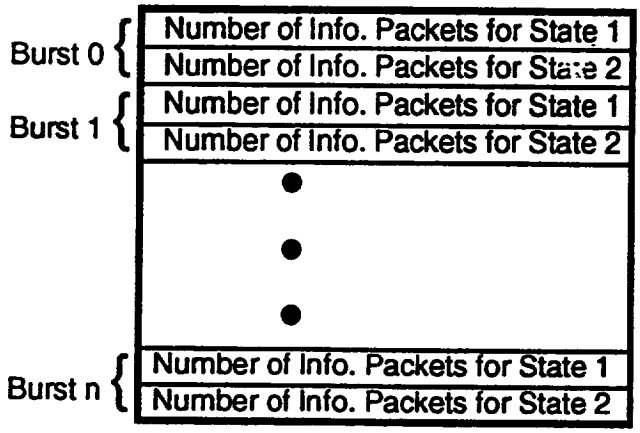

Figure 5-4. Burst Traffic Profile Format for MMDP Process

It is possible to make both the state duration and the arrival rate as a function of time. The burst traffic profile format is shown in Figure 5-5. The burst traffic profile is a series of bursts and each burst consists of four elements: the arrival rate for state 1, the number of information packets for state 1, the arrival rate for state 2, and the number of information packets for state 2. At the start, the TG generates burst 0. The TG fetches the state 1 arrival rate and uses the arrival rate to generate information packets. When the number of information packets reaches the specified number in the table, generation of information packets for state 1 stops. The TG starts generating information packets for state 2. The TG fetches the state 2 arrival rate and uses the arrival rate to generate packets. When the number of information packets reaches the specified number in the table, generation of information packets for state 2 stops. Then the TG fetches the parameters for burst 1. The procedure repeats itself. A line interface is required to synchronize the uplink transmission time of information packets, fill the gaps between two information packets with idle packets and perform FEC on the packet header.

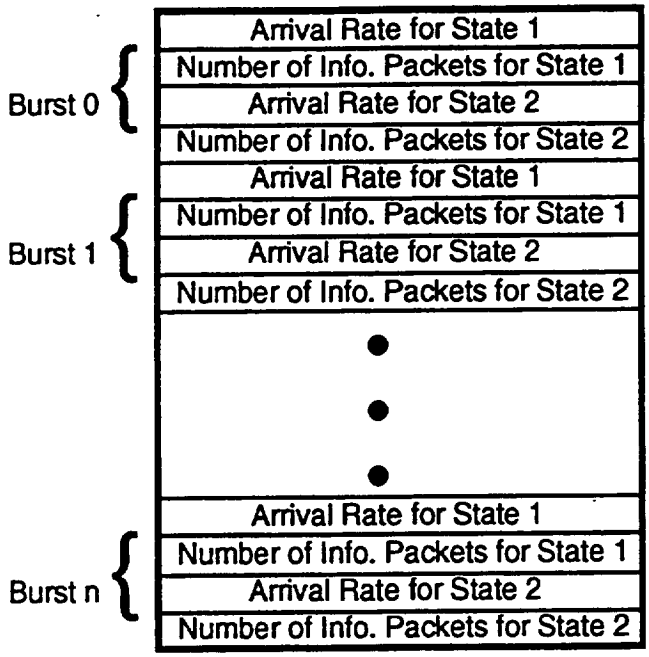

Figure 5-5. Burst Traffic Profile Format for MMDP when Parameters are Function of Time

The functional diagram of the processor board is shown in Figure 5-6. The arrival of an information packet triggers the generation of a random number. The random number is used as the input to the selection circuit to select a packet from the sample packet header pool. The testing fields such as SN and departure time stamp are generated in real time.

Although the concept of generating the MMDP packet arrivals on line can be extended to that for the MMPP packet arrivals, the MMPP packet arrivals are more difficult to generate on the fly. The reason is that the MMDP process generates packets at a constant rate while the MMPP generates packets following an exponential distribution. Constant packet arrivals can be implemented using a counter. To generate an exponential interarrival time, the processor must generate a random number and then apply the inverse transform technique [5-2]. The theory of generating a random number is presented first. Then the theory behind the inverse transform technique is described using the exponential arrival time as an example

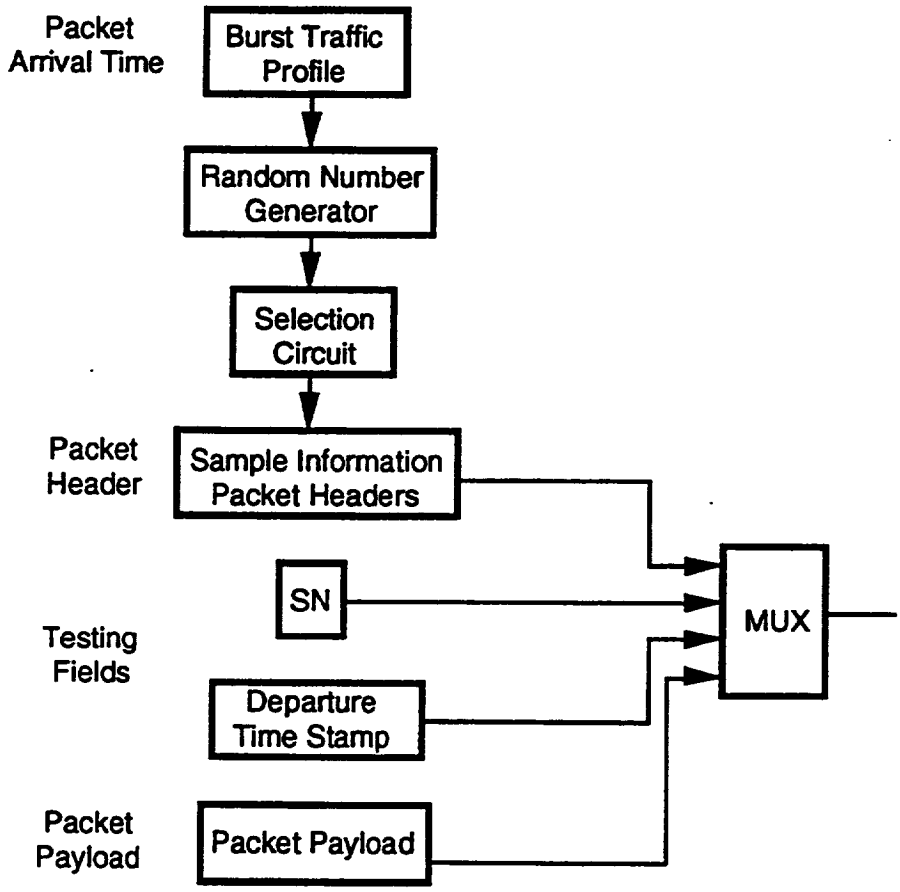

Figure 5-6. Functional Block Diagram of Processor Board

Since the random number is generated by executing an algorithm, the random numbers actually can be predicted. These random numbers are referred as "pseudo" random numbers. Most of the pseudo random numbers are uniformly distributed between 0 and 1. Since there are only a finite number of possible values for a pseudo random number (due to the number of bits used to represent a number in a machine is finite), after a certain period, the pseudo random numbers will repeat themselves. The period of the pseudo random numbers must be larger than the duration of a simulation or a test. Two algorithms are described below to generate the pseudo random numbers. The first one is the congruential generator.

$$x_i = A\, x_{i-1} + B \pmod{C}$$

$$rn = x_i/C.$$

The x_0 is the seed of the pseudo random numbers. In order for the repetition period to be very long, the A, B, and C constants must be very large. This method is more suitable for a mainframe computer where representation of a large number is an easy task.

The second algorithm was proposed by Wichmann and Hill [5-3]. The algorithm needs three seeds and uses three linear congruential generators.

$x_i = A\, x_{i-1} \pmod P$

$y_i = B\, y_{i-1} \pmod Q$

$z_i = C\, z_{i-1} \pmod R$

$rn = (x_i/P + y_i/Q + z_i/R) \pmod 1$, where P, Q, and R are distinct prime numbers and x_0, y_0, and z_0 are three seeds.

The advantage of the second algorithm is that they can be implemented on 8- or 16-bit machines.

To implement a pseudo random number generator, there are three methods. The first is to use a lookup table. A large amount of pseudo random numbers are pregenerated and stored in a table. The second is to compute a pseudo random number on line using one of the two algorithms by software. The third is to compute a pseudo random number on line using one of the two algorithms by hardware.

The inverse transform technique is described below.

The exponential distribution has the probability density function as

$$a(t) = \lambda e^{-\lambda t}, \text{ where } t > 0$$

and the probability cumulative function as

$$A(t) = 1 - e^{-\lambda t}, \text{ where } t \geq 0.$$

Let $R = A(t) = 1 - e^{-\lambda t}$. Then t can be expressed as

$$t = \frac{-1}{\lambda} \text{Log}\,(1-R).$$

The value of R (rn) is between 0 and 1 and it follows the uniform distribution. Since R (rn) and t have a one-to-one correspondence, by generating a value for rn, a value for an interarrival time t can be obtained by applying the inverse transform.

To generate a series of samples (t_i) which have an exponential distribution, the following steps should be followed.

Step 1: Generate a random number rn_i.

Step 2: Compute $t_i = \frac{-1}{\lambda} \text{Log}\,(1-rn_i)$.

Step 2 can be implemented by a lookup table, software or hardware. The burst traffic profile format for an MMPP is similar to that in Figure 5-5.

The hardware/software requirements are listed in Table 5-4.

Table 5-4. Hardware/Software Requirement for Configuration B

Hardware	Software
workstation to generate burst traffic profile and sample packets off-line	statistical on-off queueing model
high-speed bus	interface driver
processor board to generate packet arrival times based on the burst traffic profile	
processor board to select a sample packet header in the file	
processor board to generate testing fields such as SN and time stamp	
line interface to synchronize transmission time, to insert idle packets, and to perform FEC on packet header	

The configuration B traffic generator, which uses software process driven approach, is feasible for low speed applications. For high-speed TGs, there are two alternatives. The first is to use a multiplexer to multiplex the traffic generated by several low-speed TGs into a high-speed stream. For example, a TG represents a carrier. The traffic pattern for the uplink beam is obtained by multiplexing the traffic produced by thirty-two TGs. The second is to develop special hardware.

5.2 Off-line Dynamic Traffic Generation

If on-line real time traffic generation imposes a high hardware complexity to the testbed, then the off-line approach should be considered. The advantage of the off-line approach is that the processing time to generate a packet in real time is reduced to a minimal; however, a large amount of memory may be required to store the pregenerated data. The aggregate traffic pattern for each uplink beam is generated by the network simulator (using event-driven simulation techniques). The test engineer specifies the duration of the simulation time, which is equivalent to the duration of real-time testing. The network simulator stores the output of the simulation in a file. After the traffic generation is completed, the file is transferred from the network simulator to the traffic generator. When the test starts, the TG simply plays out the data in the file. The configuration for off-line traffic generation is shown in Figure 5-7.

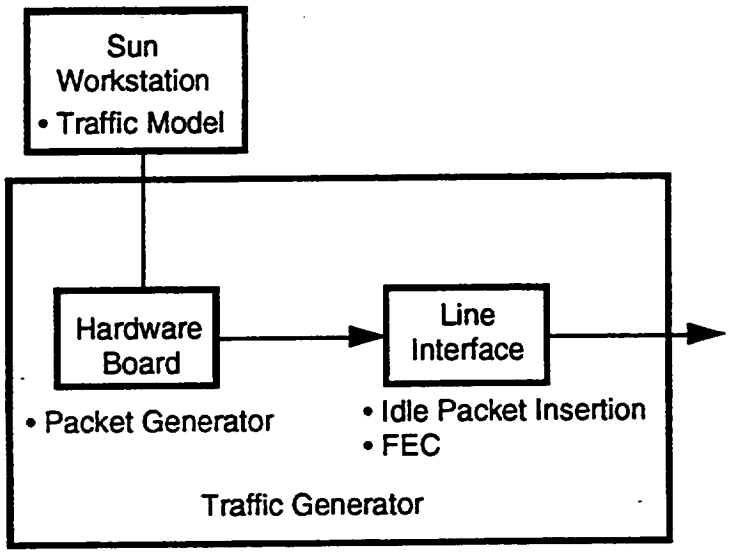

Figure 5-7. Traffic Generator Configuration C

The determination of the simulation duration is based on whether the testing is to verify the switch operation (such as routing function and congestion control algorithm) or to measure the performance. To verify the switch operation, the simulation time is not the primary concern. What is paramount is the traffic pattern. For example, to test the switch routing function, the generated traffic patterns must be able to test different switching paths of the switch. To test the effectiveness of the congestion control algorithm, the traffic pattern must be able to generate traffic overload on certain beams.

To take measurements on the switch performance (such as packet loss ratio), in addition to a realistic traffic pattern, the simulation time must be long enough such that the measurement result can achieve a certain confidence interval. The following discusses the simulation duration (or real time testing duration) required to achieve a certain confidence interval for a measured parameter.

Let Θ be the measured parameter. The $E[\Theta]$ is the estimator for Θ by simulation. An accuracy criterion ε is specified such that $E[\Theta]$ can be used to estimate Θ with the accuracy ε with a probability of $1-\alpha$. In other words,

$$P(|E[\Theta] - \Theta| < \varepsilon) \geq 1 - \alpha. \qquad (5\text{-}1)$$

Let the sample variance of Θ be S and the number of observations (or the size of the sample space) be N. If all the samples are statistically independent, the variance of the estimator $E[\Theta]$ is given as

$$\text{var}[E[\Theta]] = \frac{S}{N} = \sigma^2(E[\Theta]) = (\sigma_\theta)^2.$$

Then the statistic $t = \frac{E[\Theta] - \Theta}{(\sigma_\theta)}$ is approximately student-t distributed with a freedom of N [5-2]. Therefore, the 100 $(1 - \alpha)$% confidence interval for $E[\Theta]$ is expressed as

$$E[\Theta] - t(\alpha/2,N-1)(\sigma_\theta) \le \Theta \le E[\Theta] + t(\alpha/2,N-1)(\sigma_\theta), \quad (5\text{-}2)$$

where $t(\alpha/2,N-1)$ is defined as $P(t \ge t(\alpha/2,N-1)) = \alpha/2$.

Based on Equation 5-1, to satisfy a given accuracy ε, the following equation must hold true:

$t(\alpha/2,N-1)(\sigma_\theta) \le \varepsilon$. Substitute (σ_θ) with $\sqrt{\frac{S}{N}}$, then

$$N \ge \left(\frac{t(\alpha/2,N-1)}{\varepsilon}\right)^2 S \quad (5\text{-}3)$$

Since $t(\alpha/2,N-1) \ge z(\alpha/2)$, where $z(\alpha/2)$ is defined as $P(z \ge z(\alpha/2)) = \alpha/2$ for a normal pdf = $\frac{1}{\sqrt{2\pi}}\exp(-z^2/2)$ with mean zero and variance 1, the above equation can be written as

$$N \ge \left(\frac{z(\alpha/2)}{\varepsilon}\right)^2 S \quad (5\text{-}4)$$

Let $\varepsilon = E\Theta$. Then

$$N \ge \left(\frac{z(\alpha/2)}{E\Theta}\right)^2 S,$$

Therefore, the value of N depends on the value of S, the variance of the estimator. It was suggested in Reference 5-1 that the value of S can be in the same order as the estimated quantity Θ, when Θ is in the order of 10^{-n}, where $n \ge 1$. To estimate a packet loss ratio (Θ) of 10^{-n} with E =0.1 and 95% confidence interval (α = 0.05) or 90% confidence interval (α = 0.1), the minimum number of samples (packets) which required to be generated is listed at Tables 5-5 and 5-6.

Table 5-5. Number of Packets Required to Estimate PLR for 95% Confidence Interval

PLR	Number of Packets
10^{-6}	2.4×10^7
10^{-7}	2.4×10^8
10^{-8}	2.4×10^9

Table 5-6. Number of Packets Required to Estimate PLR for 90% Confidence Interval

PLR	Number of Packets
10^{-6}	2.3×10^7
10^{-7}	2.3×10^8
10^{-8}	2.3×10^9

Note that when the testing duration is longer than the duration of the data stored in the file, it is inevitable that the TG must repeat its traffic pattern. In order not to duplicate the traffic history, different TGs may select a starting point of the file to read differently. In this case, the interaction among different TGs on the switch will be different and the measurement results will be different from the previous ones. Now the question is what should be the format of the file. There are several alternatives.

- The first is the network simulator generates the exact output format for entire testing duration, i.e., information packets and idle packets with headers.
- The second is the network simulator only generates the headers for information packets and idle packets.
- The third is the network simulator generates the arrival times and their associated information packets.
- The fourth is the network simulator generates the the arrival times and their associated information packet headers.
- The fifth is the network simulator generates the arrival times of the information packets for the entire testing duration. A small amount of sample information packet headers are also generated.
- The sixth is the network simulator generates the busy/idle status of the traffic generator for the entire testing duration. A small amount of sample information packet headers are also generated.
- The seventh is the network simulator only generates a small amount of sample packets.

For the first four file formats, a packet header contains 1) destination field, 2) source field, 3) priority field, 4) control field, 5) FEC field, and 6) testing field. The testing field contains VCN, SN and arrival time stamp. There are two different time stamps: the arrival time and the departure time. The arrival time stamp means the time the packet arrives to the TG. The departure time stamp means the time the packet leaves the TG. The arrival time of a packet can be pregenerated and the departure time must be generated in real time. If the traffic generator has not implemented the congestion control algorithm, the arrival time is equivalent to the departure time. When the congestion control algorithm is applied, the packets may be temporally delayed at the traffic generator. In this case, the difference between the arrival time and the departure time represents the queueing delay at the traffic source during congestion.

For the fifth and the sixth file formats, a packet header contains 1) destination field, 2) source field, 3) priority field, 4) control field, 5) FEC field, and 6) testing field. The testing field contains VCN. The arrival time stamp information is available from the file and the departure time stamp needs to be generated in real time. The SN for each packet has to be generated in real time.

For the seventh file format, a packet header contains 1) destination field, 2) source field, 3) priority field, 4) control field, 5) FEC field, and 6) testing field. The testing field contains VCN. The departure time stamp and SN for each packet must be generated in real time.

The pros and cons of these output formats are addressed below.

The first output format alternative is to replicate the complete profile of the traffic pattern for the entire testing duration . An example of the file format is illustrated in Figure 5-8. After the traffic generator receives the file, the TG sends out the packets in the file using the constant source rate. The source rate is the same as the uplink rate. The functional block diagram of the hardware board for file format alternative I is shown in Figure 5-9.

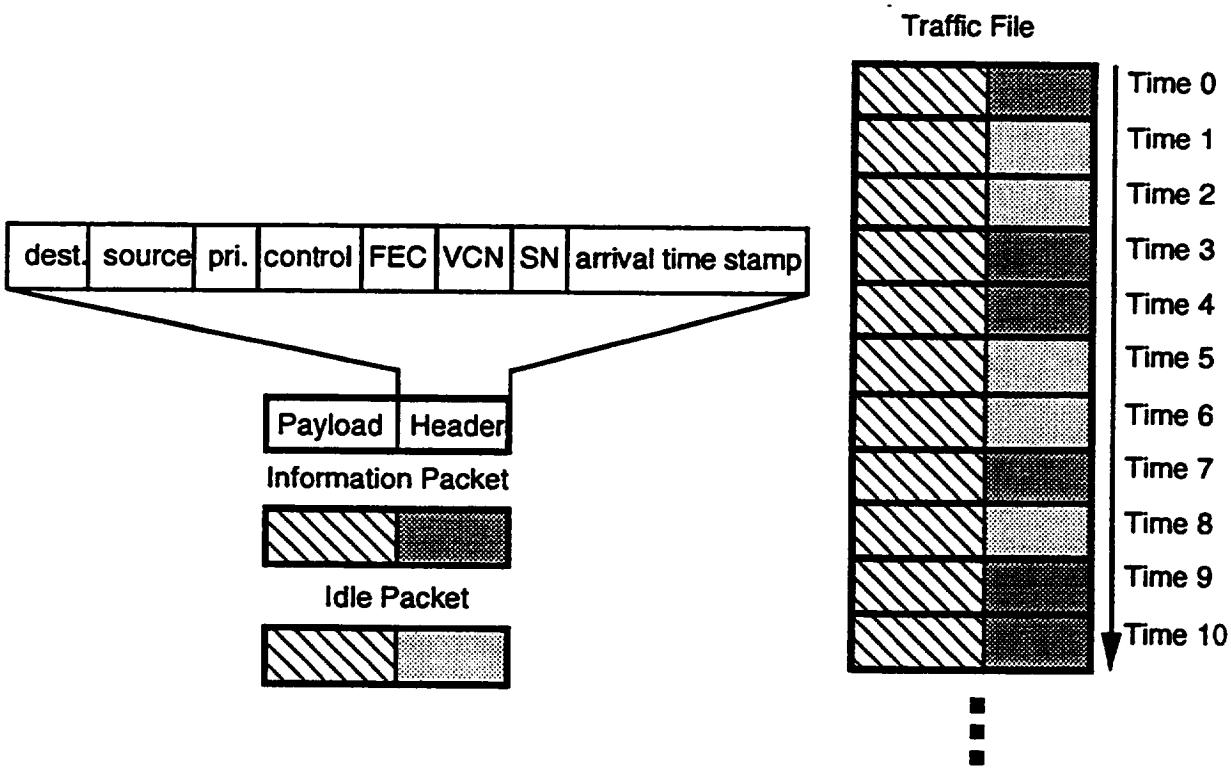

Figure 5-8. Traffic File Format Alternative I for Configuration C

The second alternative basically is the same as the first one except the traffic generator needs to generate the packet payload itself. Since the packet payload is not important in testing the switch, the payload contents can be the same for every packet or contain pseudo random sequence numbers. Therefore, it may be more advantageous to let the traffic generator to generate the packet payload such that the file storage requirement can be reduced. The functional block diagram of the hardware board for file format alternative II is shown in Figure 5-10.

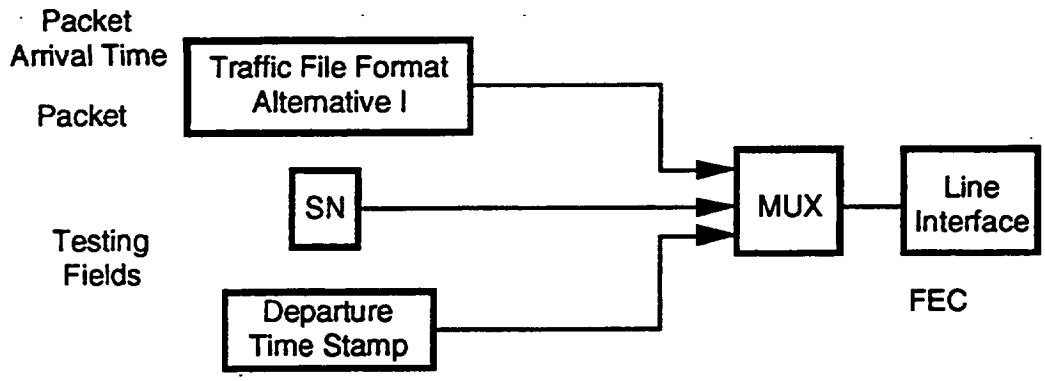

Figure 5-9. Functional Block Diagram of Hardware Board for File Format Alternative I

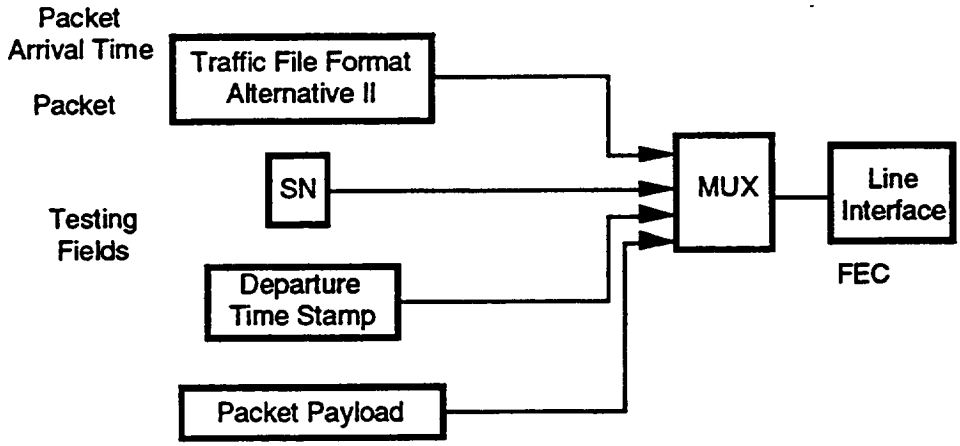

Figure 5-10. Functional Block Diagram of Hardware Board for File Format Alternative II

The third alternative only generates the information packets. To know the arrival time of each information packet, a time stamp is sent with the information packet [5-4]. The time stamp contains the arrival time of a packet. The time stamp and the corresponding packet are store in a memory. The file format is illustrated in Figure 5-11. The scheme is used in Reference 5-1, which is similar to event driven simulation. The source reads the next time stamp in memory. The time stamp is compared with the current time. The difference is used to set a timer. When the timer expires, the TG sends out the information packet. And then the same procedure repeats. The functional block diagram of the hardware board for file format alternative III is shown in Figure 5-12.

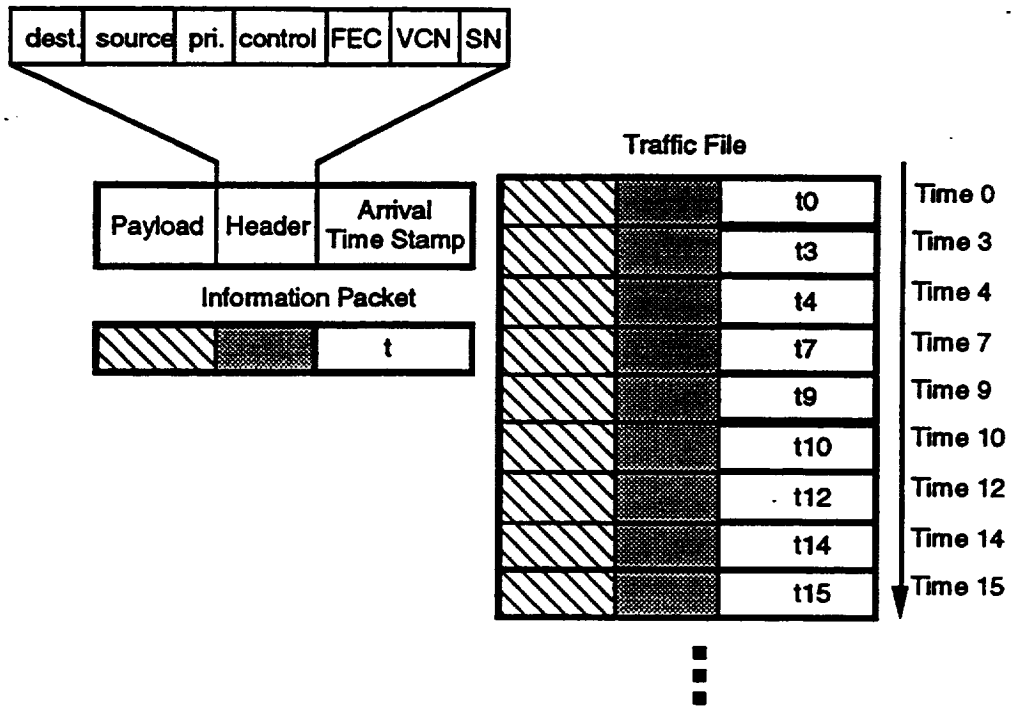

Figure 5-11. Traffic File Format Alternative III for Configuration C

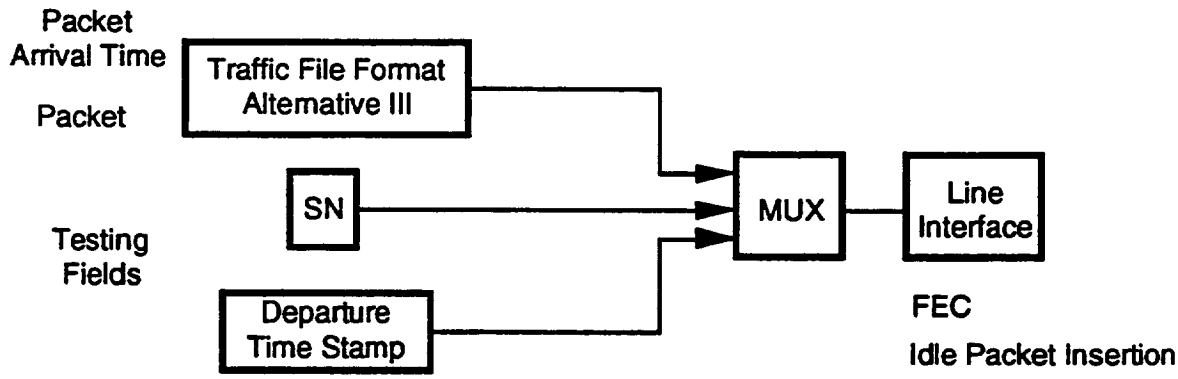

Figure 5-12. Functional Block Diagram of Hardware Board for File Format Alternative III

The fourth alternative is similar to the third alternative except only the packet headers of the information packets and their arrival times are generated. The traffic source is responsible of generating the payload. The functional block diagram of the hardware board for file format alternative IV is shown in Figure 5-13.

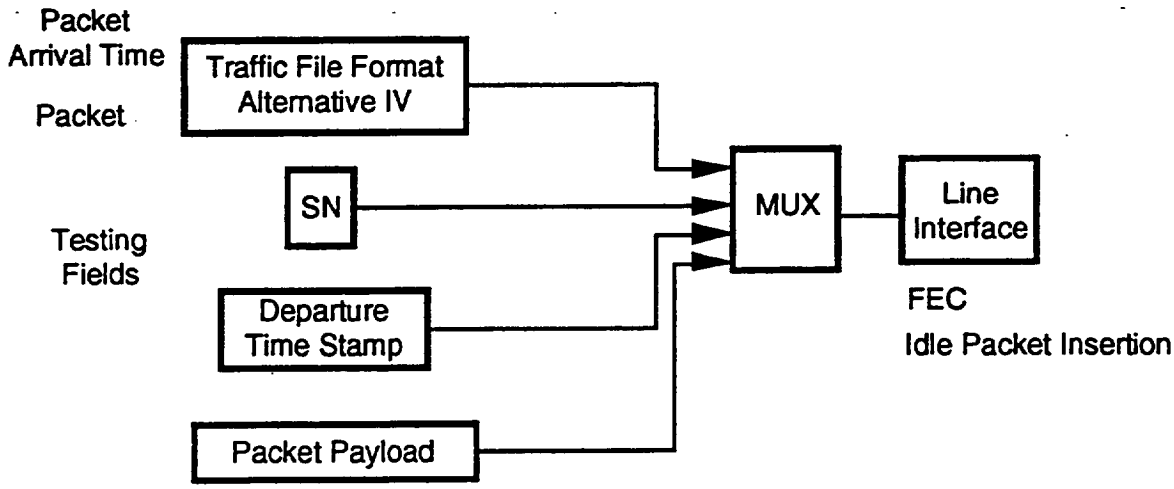

Figure 5-13. Functional Block Diagram of Hardware Board for File Format Alternative IV

The fifth alternative is to generate the arrival times of the information packets for the entire testing duration. The file format is illustrated in Figure 5-14. A pool of sample packet headers are constructed off-line. When the TG needs to generate an information packet, the TG simply picks one of the packet headers in the pool either randomly, following a specific sequence or following a certain distribution. The possible implementation for a pseudo random number generator and the inverse transform scheme has been discussed before. The arrival times can be used as the arrival time stamp. However, the TG must generate SN and departure time stamp in real time.

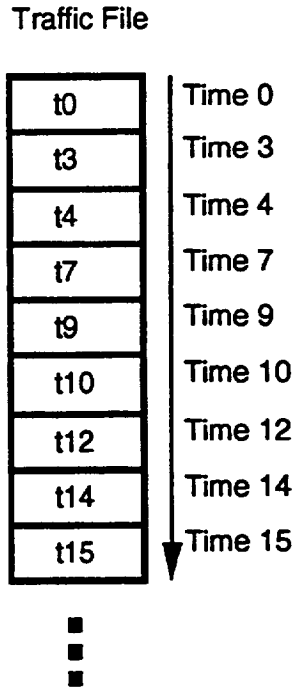

Figure 5-14. Traffic File Format Alternative V for Configuration C

The functional block diagram of the hardware board for file format alternative V is shown in Figure 5-15.

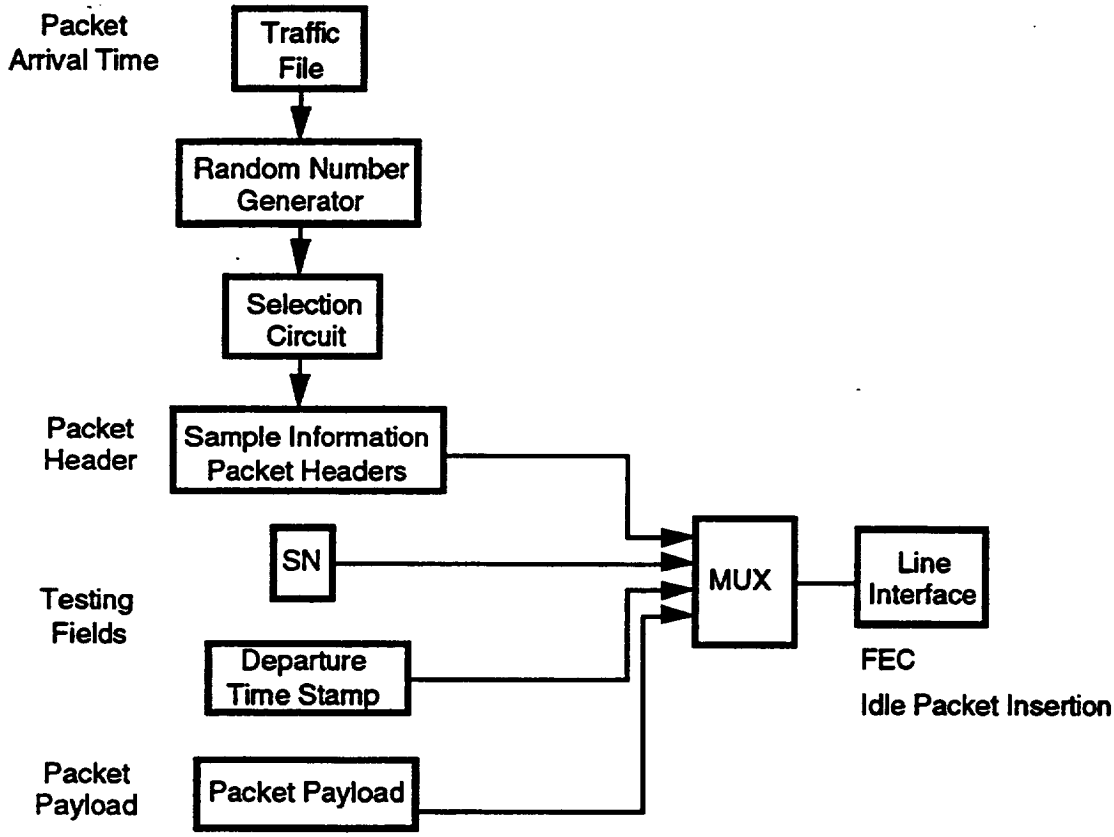

Figure 5-15. Functional Block Diagram of Hardware Board for File Format Alternative V

The sixth alternative is to generate the busy/idle status of the traffic generator for the entire testing duration. The busy/idle status can be represented using a bit. The file format is illustrated in Figure 5-16. A pool of sample packet headers are constructed off-line. The sample packet headers cover all the possible combinations of different destinations and priorities. The TG examines the busy/idle bit at every slot. If it is a busy bit, the TG simply picks one of the packet headers in the pool either randomly, following a specific sequence or following a certain distribution. If it is an idle bit, the TG generates an idle packet. The time stamp can be represented using the number of bits which have been elapsed. For example, the first bit corresponds to time 31.25 ms, the second bit corresponds to 2x31.25 ms, and so on. Since the arrival times are already available and can be used as the time stamp field, the TG only has to attach the time stamp in the packet header. However, the TG must generate the sequence number in real time. Since the file only contains the busy/idle status for the traffic generator and a small amount of sample packets, this alternative results in the lowest storage requirement compared with the previous alternatives. The functional block diagram of the hardware board for file format alternative VI is similar to that for file format alternative V and will not be repeated here.

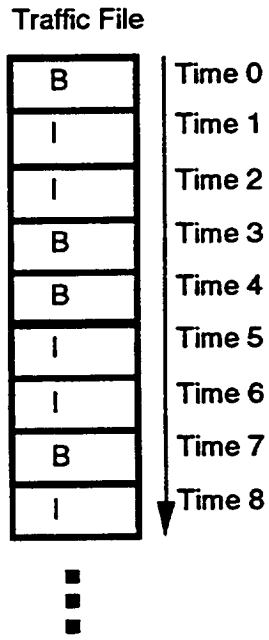

Figure 5-16. Traffic File Format Alternative VI for Configuration C

The seventh alternative is to have only a pool of sample packet headers. This alternative is the same as static testing where the packet arrival times are constant.

The following computes the storage requirement for each alternative. Assume the arrival time, which is essentially the time stamp, is one of the testing fields in the packet header. Let the smallest time unit is the packet slot time (31.25 μs). The time stamp size is 32 bits, which can be used to represent an absolute time from 31.25 μs to 134217.728 sec (or 37.28 hours). The size of the packet header is 128 bits and that of the packet is 2048 bits. Assume the measured parameter is the packet loss ratio. To estimate a packet loss ratio of 10^{-6} with an accuracy of 0.1 and 95% confidence interval, the number of packets required to be generated is at least 2.4×10^7. Since there are eight TGs, the number of packets required to be generated for one TG is 3.0×10^6, which is equivalent to 93.75 sec if packets are generated using the link rate. If the link utilization is 0.9, then the number of packets required to be generated will be 3.33×10^6, where there are 3.0×10^6 information packets and 0.33×10^6 idle packets.

For file format alternative I, the memory has to store the information packets and idle packets for the entire testing duration. Assume the link utilization is 0.9. The amount of storage is 6.82×10^9 bits, which is equivalent to 852.5 Mbytes. If the storage requirement is too large for RAMs, then disk or tape may be considered. Assume the bus width of the memory is 32 bits. The memory access time requirement is ≤ 0.488 μs since a 2048-bit packet must be read out in less than 31.25 μs.

For file format alternative II, the memory has to store the information packet headers and idle packet headers for the entire testing duration. Assume the link utilization is 0.9 The amount of storage is 4.26×10^8 bits, which is equivalent to 53.25 Mbytes.

Assume the bus width of the memory is 32 bits. The memory access time requirement is ≤ 7.8125 μs since a 128-bit header must be read out in less than 31.25 μs.

For file format alternative III, the memory has to store the arrival times and their associated information packets for the entire testing duration. Note the arrival times for idle packets are not stored. Assume the link utilization is 0.9. The amount of storage for information packets is 6.144×10^9 bits, which is equivalent to 768 Mbytes. Assume the arrival time is one of the header fields such that the arrival times will not consume additional memory. Assume the bus width of the memory is 32 bits. The memory access time requirement is ≤ 0.488 μs.

For file format alternative IV, the memory has to store the arrival times and their associated packet headers for the entire testing duration. Assume the link utilization is 0.9. The amount of storage for information packet headers is 3.84×10^8 bits, which is equivalent to 48 Mbytes. Assume the arrival time is one of the header fields such that the arrival times will not consume additional memory. Assume the bus width of the memory is 32 bits. The memory access time requirement is ≤ 7.8125 μs.

For file format alternative V, the memory stores the arrival times for the entire testing duration and a pool of sample packets. Assume the link utilization is 0.9. The amount of storage for arrival times is 9.6×10^7 bits, which is equivalent to 12 Mbytes. Assume the sample packets consist of packets directed to different beams (both point-to-point and multicast), to different dwells (point-to-point), and of two priorities. There are 4096 combinations; each combination corresponds to a unique VCN. Since the packet header has 128 bits, the additional amount of storage is 0.065 Mbytes. Assume the bus width of the memory is 32 bits. The memory access time requirement is ≤ 6.25 μs.

For file format alternative VI, the memory stores the busy/idle status of the traffic generator for the entire testing duration and a pool of sample packets. The amount of storage for arrival times is 3.3×10^6 bits, which is equivalent to 0.4125 Mbytes. The additional amount of storage for packet header is the same as file format V, which is 0.065 Mbytes. Assume the bus width of the memory is 32 bits. The memory access time requirement is ≤ 7.8125 μs.

For file format alternative VII, the memory stores a pool of sample packets. Assume the sample packets consists of packets directed to different beams (both point-to-point and multicast), to different dwells (point-to-point), and of two priorities. There are 4096 combinations. Since the packet header has 128 bits, the additional amount of storage is 0.065 Mbytes. Assume the bus width of the memory is 32 bits. The memory access time requirement is ≤ 7.8125 μs.

From the above comparisons, file format IV does not save a significant amount of memory compared with file format II. The only difference between file format IV and file format II is that format IV does not save the arrival times of idle packets in the file and format II does. However, the hardware complexity for file format IV is expected to be higher than file format II because the packets in file format II can be read out using a simple counter while the packets in file format IV needs to use a timer to be generated.

File formats I and II require the least hardware and file format VI has the least amount of memory requirement compared with the other (dynamic) file formats. A comparison of the storage and memory access time requirements for different file formats is provide in Table 5-7. The hardware/software requirements for the off-line approach are listed in Table 5-8.

Table 5-7. Storage and Memory Access Time Requirements for Different File Formats

Format	Storage (Mbytes)	Memory Access Time (μs)	Remark
I	852.5	0.488	
II	53.25	7.8125	need to attach packet payload
III	768	0.488	0.9 link utilization, 32 bit time stamp
IV	48	7.8125	0.9 link utilization, 32 bit time stamp, need to attach packet payload
V	12+0.065	6.25	beam multicast, 2 priority, 32 bit time stamp, need to attach packet payload
VI	0.4125+0.065	7.8125	busy/idle status, time stamp is abstract,beam multicast, 2 priority, need to attach packet payload
VII	0.065	7.8125	static testing, beam multicast, 2 priority, need to attach packet payload

Table 5-8. Hardware/Software Requirements for Approach D

Hardware	Software
workstation to generate the busy/idle status of the traffic generator for the testing duration	statistical queueing model
high-speed bus	interface driver
memory to store the traffic file	algorithm of selecting a sample packet header in the file
hardware board to generate a packet arrival time based on the traffic file, to read a packet header from the memory, to attach the packet payload, and to generate SN and departure time stamp	
line interface to synchronize transmission time, to insert idle packets, and to perform FEC on packet header	

The necessary modifications of the TG to provide the traffic source response procedure to congestion are discussed below.

5.3 Dynamic Traffic Generation Considering Congestion Control

In order to determine the congestion response procedure in the TG, the congestion control performance, the congestion detection procedure in the switch, and the monitoring capability of the testbed must be defined first.

5.3.1 Congestion Control Performance

The congestion control performance can be evaluated by

1. on-board packet loss ratio under normal and overload conditions
2. downlink beam throughput under normal and overload conditions
3. end-to-end packet delay under normal and overload conditions
4. congestion detection time
5. congestion response time
6. congestion recovery time.

An effective congestion control algorithm should maintain the PLR at the desired levels on-board under different traffic loading conditions. A congestion control algorithm with an extremely low PLR (due to overreacting to congestion at the traffic source) or a high PLR (due to slow response to congestion at the traffic source) is considered to be inefficient. An effective congestion control algorithm should also maximize the satellite resource, i.e., the downlink beam throughput. When congestion occurs, the (data) packets are buffered at the source. The packet loss is reduced at the expense of the packet delay. The end-to-end delay can be used to identify the congestion interval and the protocol parameters (such as the time-out value). The congestion detection time is defined as the interval between the time of congestion onset at the traffic source and that of congestion detection at the satellite. The congestion response time is defined as the interval between the time of congestion detection at the satellite and that of congestion relief at the satellite. The congestion recovery time is defined as the interval between the end of overload period at the source and the instant that the end-to-end delay is back to normal [5-5].

The long-term average measurements can not catch the dynamic behavior of the algorithm. The PLR, downlink beam throughput, and the end-to-end delay should be measured as a function of time.

5.3.2 Congestion Detection Procedure

Two switching architectures: shared memory switch and shared bus switch were selected as candidates in Reference 1-1 for subsequent breadboard design. The proposed

congestion control schemes in Reference 4-1 are directly applicable to these two switching architectures. The radical preventive congestion control scheme (A6) is selected as the representative for discussion since it has the simplest implementation and it is well suited for the meshed VSAT satellite environment. However, the discussion of implementing the source response procedure to congestion is general enough such that it can be extended to other schemes easily. Although the congestion control scheme only deals with beam overload, the scheme can be directly extended to dwell overload. The congestion control scheme comprises two procedures: congestion detection procedure and source response procedure to congestion. The congestion detection procedure is discussed below.

Congestion is detected when the switch loading exceeds a certain threshold. As illustrated in Figure 5-17, the switch loading can be represented by

- mean arrival loading to the queue.
- queue output utilization
- average queue length.

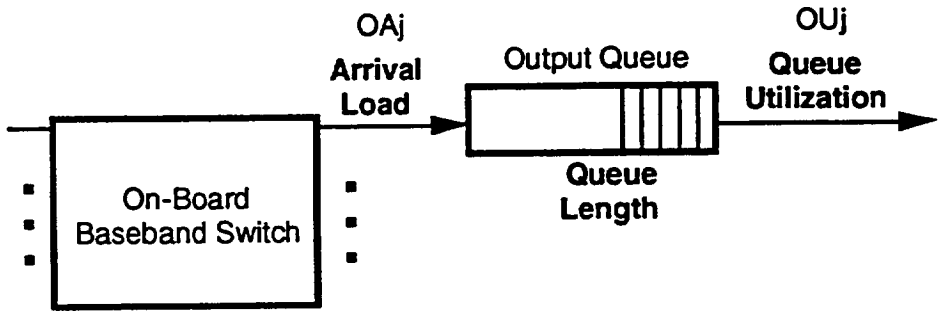

Figure 5-17. Representations of Switch Loading

Define the following notation:

OA: the mean arrival loading of the switch.

OA_j: the mean arrival loading at output j, where $0 \leq OA_j < 8$.

OU_j: the mean queue utilization at output j, where $0 \leq OU_j < 1$.

oa_{ij}: the mean arrival loading at output j from input i, where $0 \leq oa_{ij} < 1$.

ou_{ij}: the mean queue utilization at output j from input i, where $0 \leq ou_{ij} < 1$.

The mean output queue arrival loading (OA_j) is defined as

$$\frac{\text{number of packets in measurement interval}}{\text{number of uplink slots in measurement interval}}$$

where an uplink slot is defined as packet size/uplink speed.

The mean output queue utilization (OU_j), which is also the mean downlink beam utilization, is defined as

$$\frac{\text{number of packets in measurement interval}}{\text{number of downlink slots in measurement interval}}$$

where a downlink slot is defined as packet size/downlink speed.

The congestion detection procedure for different switch loading representations is described in the following.

5.3.2.1 Mean Arrival Loading

The mean arrival loading is measured in front of the queue. The mean output queue arrival loading (OA_j) measures the traffic multiplexed from different uplink beams destined to downlink beam j. In general, $0 \leq$ mean arrival loading (OA_j) $< N$ for output buffering.

The summation of the mean arrival loading to output j from different inputs divided by 8 is the mean arrival loading to output j.

$$OA_j = (\sum_{i=0}^{7} oa_{ij})/8.$$

The summation of the mean arrival loading at different output ports divided by 8 is equal to the average output arrival loading, i.e.,

$$OA = (\sum_{i=0}^{7} OA_i)/8.$$

The mean arrival loading at the output port is limited by the switch throughput and the downlink capacity. Clearly, the switch throughput must be larger than the downlink capacity for output queueing to occur. From the M/D/1 queueing theory, the mean arrival loading should not exceed (0.9 x downlink capacity); otherwise, the queueing delay at the output queue will be increased exponentially.

Depending on the degree of fairness (and the monitoring capability) considered, more measurements may be required (on the satellite and/or the ground). A fair congestion control algorithm will reduce the user traffic volume based on the percentage the user contributes to congestion. The higher degree of fairness, the more of the measurement

results and the more complex of the algorithm. As illustrated in Figure 5-18, the measurement results of traffic loading can range from average output loading, average individual output loading, average individual output loading from different uplink beams, average individual output loading from different terminals, and average individual output loading from different virtual circuits.

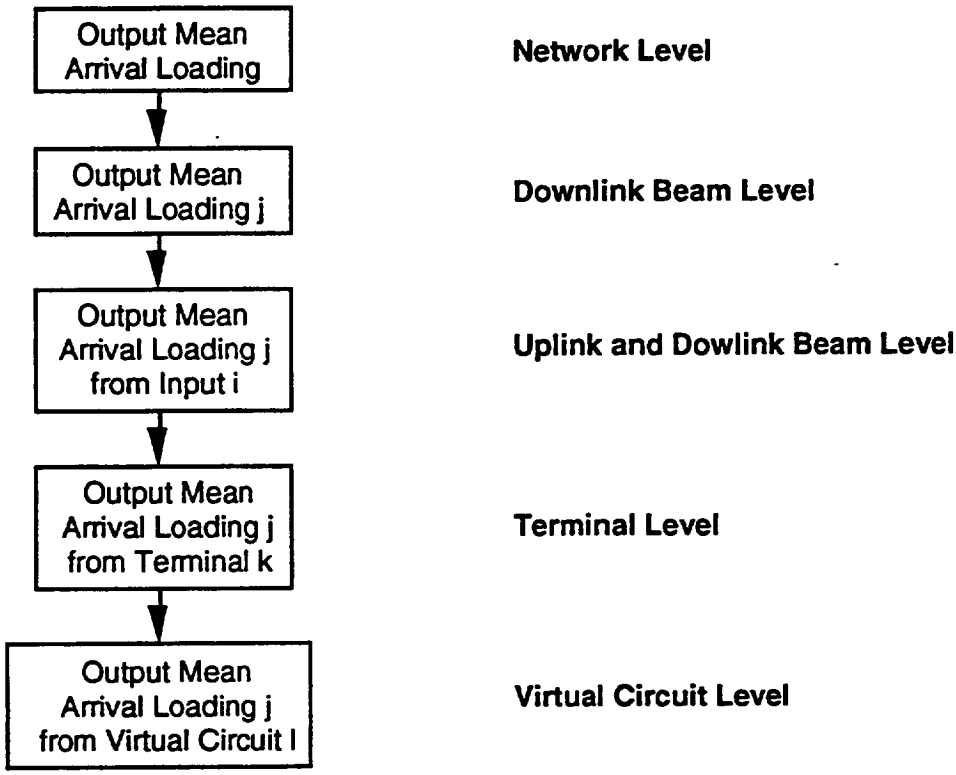

Figure 5-18. Different Levels of Congestion Control

By measuring the average output arrival loading OA, the amount of traffic reduction for the whole network can be determined. An operation range, (OA_l, OA_h), is established for OA. The midpoint of the operation range is denoted as OA_m. The operation range is obtained from the PLR curve derived from experiments, simulation, or queueing analysis. If $OA > OA_m$, congestion is detected. A congestion control message is broadcast to different beams, i.e., the OBSC sends out congestion control messages to all terminals. The amount of traffic reduction, which should be undertaken by all terminals is $\frac{OA_m}{OA}$. The traffic reduction algorithm uses the centralized scheme since the amount of traffic reduction is instructed by the OBSC.

By measuring the OA_j, the degree of congestion at different downlink beams can be identified. To guarantee the QOS of different connections on-board, an operation range, (OA_{jl}, OA_{jh}), is established for each OA_j. The midpoint of the operation range is denoted as OA_{jm}. The operation range is obtained from the PLR curve derived from experiments, simulation, or queueing analysis. If $OA_j > OA_{jm}$, congestion is detected. A congestion control message is broadcast to different beams, i.e., the OBSC sends out

congestion control messages to all terminals. The amount of traffic reduction, which should be undertaken by all terminals, for downlink beam j is $\frac{OA_{jm}}{OA_j}$. The traffic reduction algorithm uses the centralized scheme since the amount of traffic reduction is instructed by the OBSC.

The second is to measure OA_j and oa_{ij}. By measuring OA_j and oa_{ij}, not only the degree of congestion at different downlink beams can be identified, the amount of loading contributed by each uplink beam to the overload downlink beam can also be identified. In this case, the amount of traffic reduction for terminals at different uplink beams will be different based on the comparison results. As discussed before, if $OA_j > OA_{jm}$, congestion is detected. When congestion is detected, oa_{ij} is compared with oa_{ijm}. The comparison results determine the amount of traffic reduction for downlink j, which should be undertaken by terminals at uplink i. The OBSC sends out 8 "different" congestion control messages to 8 different uplink beams. The terminals at uplink i will reduce their traffic directed to downlink beam j according to the received message.

By measuring OA, the amount of traffic reduction for the network can be determined. By measuring OA_j, the amount of traffic reduction for a particular downlink beam for all terminals can be determined. In this case, all terminals in the network are required to reduce the same amount of traffic. Clearly, this is only fair if it is a homogeneous network, i.e., all the terminals in the network have exactly the same behavior. By measuring OA_j and oa_{ij}, the amount of traffic reduction for a particular downlink beam for all terminals in a particular uplink beam can be identified. In this case, all the terminals in the same uplink beam are required to reduce the same amount of traffic. This implies that the terminals in the same uplink beam are homogeneous. This assumption matches with the that made for the TG. By measuring mean output arrival loading from each terminal, the amount of traffic reduction for a particular downlink beam for a terminal can be determined. Note the amount of traffic reduction or increase will be different for different terminals. The loading from each terminal and the loading from each virtual circuit do not have to be measured at the on-board switch, they may be measured at the local terminal by monitoring the past traffic volume. In a sense, distributed processing is applied to reduce the on-board processor complexity. When the terminal receives the congestion control message, the terminal uses its own measurements to determine the amount of traffic reduction for itself or each virtual circuit.

5.3.2.2 Average Queue Length

Congestion control is performed by monitoring the output queue length. A threshold value is established for the average queue length. The threshold value is used to maintain the system PLR around the desired point (for example 10^{-9}). The threshold value should be obtained from the PLR curve derived from simulation, experiments, or queueing analysis. If the average queue length exceeds the threshold value, congestion is detected. Congestion indication message is sent to the user terminals. When the user terminal validates that the switch is indeed in congestion, the user terminals applies its own traffic reduction algorithm to reduce its own traffic. Since the user terminals apply their own algorithm for traffic reduction, it is a decentralized scheme.

5.3.2.3 Mean Queue Utilization

The mean output queue utilization (i.e., the downlink beam mean utilization) can also be used for congestion detection. In general, $0 \leq$ mean queue utilization < 1. The discussion used for the mean arrival loading is also applicable for mean queue utilization.

5.3.2.4 Implementation Considerations

To measure the mean arrival loading (OA_j), a counter is required at each output port to count the number of arrival packets. The measurement technique may use jumping window or sliding window. To measure the mean arrival loading (oa_{ij}), eight more counters are placed at the output port to count the number of packets arrived from different uplink beams. Since the packet header only contains the output port information, the input port address must be included in the packet to make measurements of oa_{ij} possible. The other alternative is to place the counters at the input port.

To measure the average queue length, the queue length at each output port must be monitored. The queue length measurement technique may use jumping window or sliding window. The arrival loading can be derived by monitoring the queue length (and the packet loss ratio). Define the notation below.

- OA_rate (t): the average arrival rate (pkts/sec) for a queue at time t.
- OU_rate (t): the service rate (pkts/sec) for a queue at time t.
- q(t): the queue length at time t.

By collecting q(t) at different instants, the average arrival rate can be computed.

$$OA\_rate(t)\,\Delta t + q(t) = OU\_rate(t)\,\Delta t + q(t+\Delta t) + \text{number of loss packets.}$$

Since OU_rate(t) and Δt are known quantities, the OA_rate(t) can be computed easily. To implement a flexible testbed which can be used to evaluate different congestion detection procedures, this method should be considered. The queue length monitoring result (and the packet loss count) can be fed into a remote processor. Then the remote processor compute the average queue length and arrival loading (OA_j).

5.3.3 Traffic Source Response Procedure to Congestion

In Reference 4-1, it was assumed that only data traffic is subject to flow control, not voice and video traffic, since voice and video traffic use dedicated carriers and a separate on-board circuit switch. In this study, an MF-TDMA access scheme and an FPS are used to provide integrated services for data, voice and video. Clearly an integrated network must guarantee the QOS for different types of traffic. For example, the packet delay jitter should be small for voice. For data, the packet loss of the loss-sensitive data

must be minimized to prevent retransmission. Two bandwidth allocation schemes are possible for the voice traffic. The first scheme is to allocate a fixed number of slots for each voice request and the voice capacity can not be shared by data traffic. The second scheme is the amount of capacity allocated to each terminal is based on the average aggregate traffic volume. An important issue of the second scheme is whether the voice packets should be subject to rate-based control. This depends on the specific capacity request/allocation scheme and voice encoding scheme. In this study, it is assumed that the voice and data statistically share the bandwidth (for a downlink beam) at the terminal and the voice is subject to rate-based control.

The capacity is allocated using demand assignment multiple access. The terminal sends a capacity request to the OBSC or NCC to set up a semi-permanent pipe from the terminal to the satellite. In this study, it is assumed that the semi-permanent pipe capacity established is fixed throughout the testing. It is further assumed that the aggregate traffic generated by different sources will never generate packets more than the capacity of the semi-permanent pipe. Network congestion is implemented by forcing the source rates at the terminals approaching, not exceeding the semi-permanent pipe capacity

The source response procedure to congestion contains four elements: filter, rate-based control, traffic reduction algorithm, and traffic increase algorithm. To perceive and test the effectiveness of the congestion control algorithm, it is inevitable that part of the functionalities of these four elements must be incorporated into the hardware design of a TG. In a sense, the traffic generator has the functions of both traffic source and terminal. The filter is to reduce the switch loading oscillation by using various methods such as validating the congestion messages, increasing the congestion detection time, or increasing the source response time to congestion. If oscillation is not a major concern, the filter can be ignored. In this case, the traffic generator responds to every control message received from the satellite. The rate-based control is to regulate the amount of traffic entering the network. When the TG receives a congestion message, the rate-based control will increase or reduce the rate accordingly by applying the traffic reduction or increase algorithms.

Conceptually, the TG consists of traffic source, rate-based control, and line interface (see Figure 5-19). The traffic source generates information packets based on some statistical queueing models. The information packets are stored in the rate-based control queue. The rate-based control in the TG generates information packet transmission time events. When the event comes, an information packet is sent out. The line interface synchronizes the packet transmission time and inserts idle packets in the packet stream when there is no packet to send.

The uplink transmission rate is a constant. The information transmission rate is adaptable based on the on-board switch loading status. The source rate is either a constant (for example voice) or adjustable (for example file transfer or variable bit rate video) using flow control. The flow control scheme is used to regulate the source rate.

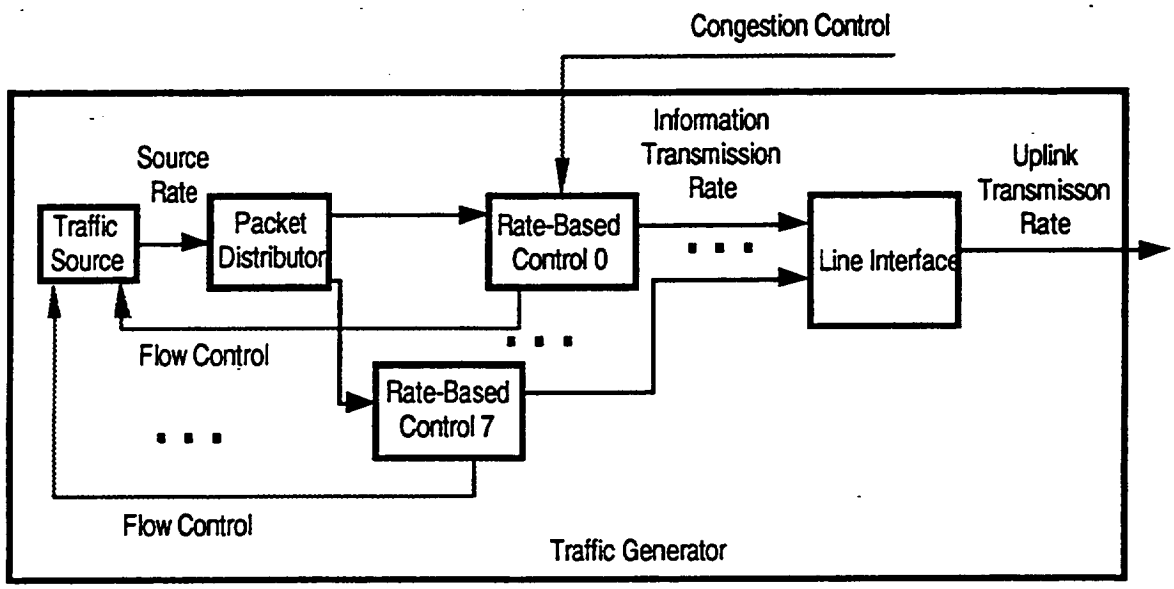

Figure 5-19. Configuration of Traffic Generator Incorporating Congestion Control

As discussed previously, the rate-based control may be available on a per satellite basis, per beam basis, or per dwell basis. If the rate-based control is available on a per satellite basis, congestion control for the switch is equivalent to congestion control of a statistical multiplexer. The switch loading is an average of eight output arrival loadings. If the rate-based control goes down to the downlink beam level, the switch loading is a vector of eight output arrival loadings. If the rate-based control goes down to the dwell level, the switch loading is a vector of sixty-four dwell loadings. In this study, it is assumed that the rate-based control is available on a per beam basis. There are eight rate-based control in a traffic generator.

There are several alternatives to handle the packets at the rate-based control in case of congestion. If the traffic source is flow controllable, the terminal should execute flow control to slow down the traffic. (If the terminal is a gateway itself, the terminal can reroute the traffic either through another satellite or using terrestrial lines.) If the rate-based control has buffering capability, the excess packets are queued; otherwise, the excess packets are dropped. Priority control may be used to guarantee the QOS of the high-priority packets at the expense of the low-priority packets. Low-priority packets are dropped before high-priority packets in case of congestion. Since dropping of data packet will result in retransmission, the data is assumed to have a higher loss priority than the voice. On the other hand, since voice can only tolerate a few hundred ms delay, the voice has a higher delay priority than data. One possible tradeoff is to design a priority control scheme at the user terminal such that the voice can have lower delay but may suffer higher packet loss ratio.

To minimize the queueing delay of the voice packet, the voice packets are served before the data packets. It is assumed that voice and data are stored in two separate virtual queues at the terminal. However, the summation of the length of the two virtual queues is a constant. The voice packets and the data packets are competing for the buffer space. To keep the PLR of the data packets low and at the same time to accept as many

voice packets as possible, there are three possible schemes for packet insertion to the queue.

1) The number of voice packets in the virtual queue is checked. If the number is above a threshold, the new arrival voice packets are dropped.

2) The summation of the length of the two virtual queues is checked. If the sum exceeds a threshold, the new-arrival voice packet is dropped.

3) The data and voice packets are accepted into the virtual queues when the buffer space is not full. When the buffer space is full, a newly arriving data packet can push the newest voice packet out of the virtual queue.

In summary, five actions are possible for the TG when the switch is congested. They are

- reduce the information rate
- reduce the source rate (by performing flow control or dynamic source coding rate adjustment).
- queue the data packet (applying priority control)
- drop the voice packets (applying priority control)
- adjust the multiplexing (service) sequence for packets with different priorities

The ideal combination is to reduce the information rate and the source rate at the same time; as a result, no packet loss will be experienced. However, the back pressure scheme for data is dependent on the end-to-end transport layer protocol and the adaptability of the source coding rate for voice and video may not be available.

Previously both on-line and off-line implementation of dynamic traffic generation are discussed. The necessary modification of the TG to incorporate congestion control is discussed below.

5.3.3.1 On-line Dynamic Traffic Generation Considering Congestion Control

In configuration A, since the workstation generates the packets on line and in real time, the congestion control message can be fed into the workstation directly (see Figure 5-20). When the workstation receives the message, the workstation applies the traffic reduction algorithm and computes a new set of parameter values for the packet arrival process in real time. This may place a large burden on the workstation since it not only

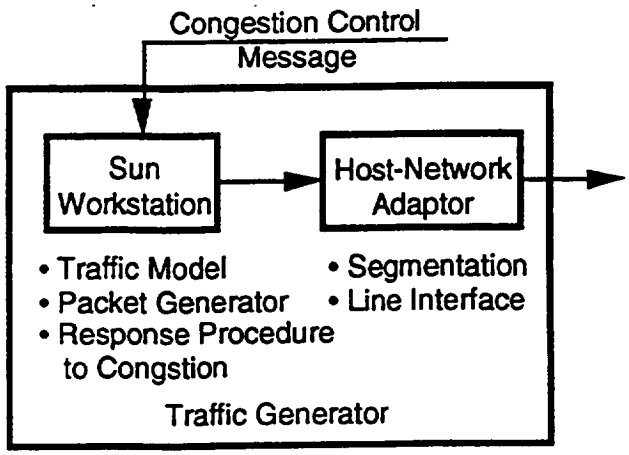

Figure 5-20. Traffic Generation Configuration A Considering Congestion Control Alternative I

has to generate packets in real time, it but also has to respond to the congestion control message and compute a new set of parameter values for the arrival process in real time.

The other alternative is to separate the traffic generation and congestion control into two entities. This configuration is shown in Figure 5-21. The implementation of the rate-based control and the synchronization function of the line interface will be discussed in the next subsection.

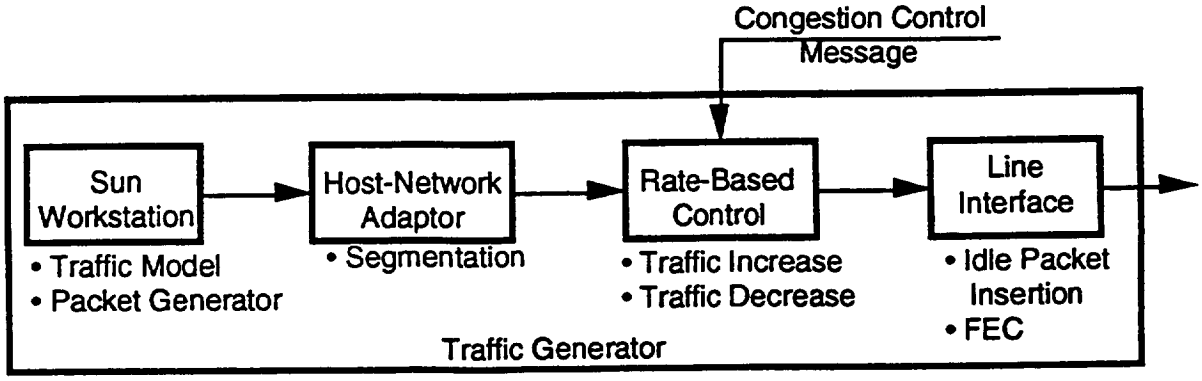

Figure 5-21. Traffic Generation Approach A Considering Congestion Control Alternative II

In configuration B, the burst traffic profile format is computed in advance. When congestion message is received, the parameter value of the burst traffic profile format must be updated. One possible implementation is described as follows. There is an offset register used to dynamically adjust the number of idle packets transmitted based on the switch loading status. When there is no congestion, the offset value is zero. The processor generates the number of idle packets based on the burst traffic profile. When the switch is congested, the processor board not only has to generate the number of idle packets specified in the burst traffic profile, it but also has to generate the number of idle packets specified in the offset register. The configuration is shown in Figure 5-22.

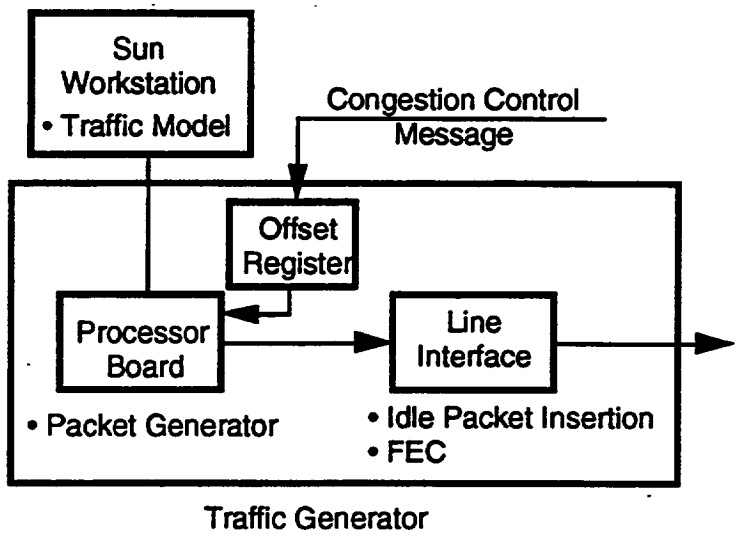

Figure 5-22. Traffic Generation Configuration B Considering Congestion Control Alternative I

The other alternative is to separate the traffic generation and the source response procedure to congestion into two different entities. This configuration is shown in Figure 5-23. The implementation of the rate-based control and the synchronization function of the line card will be discussed in the next subsection.

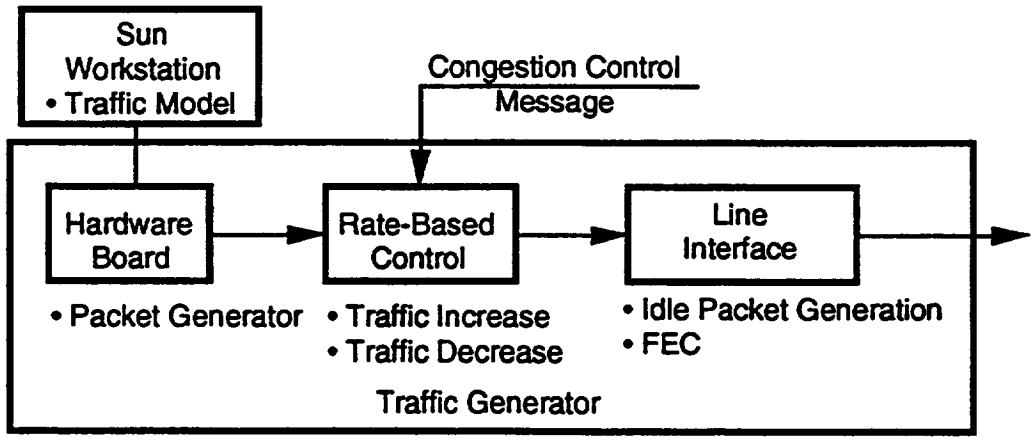

Figure 5-23. Traffic Generation Configuration B Considering Congestion Control Alternative II

5.3.3.2 Off-line Dynamic Traffic Generator Considering Congestion Control

Remember that the file format alternative I is to replicate the complete profile of the traffic pattern. The TG sends out the packets in the file using the uplink rate. There are eight rate-based control, one for each downlink beam. At the rate-based control, the packets are sent out using the assigned information transmission rate. The rate-based control is responsible for rate reduction, packet queueing, packet dropping, and rate increase. When the network is not congested, the summation of the information

transmission rates of different rate-based control is the same as the uplink transmission rate. During the congestion period, the TG will execute the congestion response procedure; as a result, the information transmission rate for the particular downlink beam is reduced. Since the rate-based control sends out the information packets in a reduced rate, additional idle packets are generated to fill the gaps in the uplink packet stream. This is performed by the line interface. Since the rate-based control regulates the rate of the information packets, the idle packets in the source file are not important. The idle packets are used to maintain the stochastic nature of the arrival process.

The implementation of different components in the TG is discussed using the enqueue procedure and dequeue procedure. The enqueue procedure is referred to the insertion of a packet into the proper location of a queue. The dequeue procedure is referred to the removal of a packet from the head of the queue. Since the traffic file of the source at the TG is transferred from the network simulator, the traffic source only has the dequeue procedure. The source clocks out the information packets in the queue (file) using the uplink rate. Since the line interface will insert idle packets in the packet stream, the idle packet is destroyed using a killing circuit before the idle packet enters the rate-based control. The dequeue procedure implementation is very simple: a RAM and a counter (to generate the address for read).

The packets coming from the source are distributed to the proper rate-based control based on the downlink beam ID. There is one queue in the rate-based control. The queue is shared by voice and data packets. The queue is implemented using two link lists, one link list is for one traffic type. The rate-based control has both enqueue and dequeue procedures. The enqueue procedure is to insert the packet at the end of link list based on the traffic type. When the queue is full, the newly arrival packet is dropped. Since the voice packet has a delay constraint, the rate-based control always sends out the voice packet first, if there is any. If no voice packets are in the queue, the rate-based control sends out the data packet instead. To bound the voice delay, the number of voice packets can be stored in the queue should be much smaller than that for data. In this study, it is assumed when the number of voice packets exceeds a certain threshold, the new arrival voice packet is dropped. (In real operation, a selective voice packet dropping strategy should be used. The selection depends on the source [variable bit-rate] coding technique, use of digital speech interpolation, and the voice regeneration protocol [recovery from packet loss]).

However, giving high priority to voice all the time may cause severe performance degradation of the data traffic if the voice traffic occupies the capacity (of the semi-permanent pipe) for a long period of time [3-21]. One alternative is to divide the capacity between voice and data, i.e., capacity allocation or bandwidth scheduling using rate-based service discipline. Rate-based service discipline is to allocate a minimum amount of capacity to one traffic type regardless of the characteristics of other traffic types [5-6]. Capacity allocation can be achieved using circular scan (or round robin) method. If voice and data have the same priority, the scanning sequence will be circular such that voice and data virtual queues can be visited alternatively. If voice and data have different priorities, weighted round robin method should be used. For example, the voice virtual queue is examined before the data virtual queue in the first M_1 visits, and the order is reversed for the next M_2 visits. After this, the routine repeats itself. One scan cycle consists of $M_1 + M_2$ visits. Another similar example is to use the (T_1,T_2)

scheme [5-7]. The voice virtual queue is served for a T_1 ms or until the voice virtual queue is empty. Then the data virtual queue is served in the same manner for a T_2 ms. Note that if there is no more packet in the virtual queue during the service period, the server always switches over to the other virtual queue. In other words, the server is never idle as long as there are packets in the virtual queue. This is also referred as work-conserving discipline, which should be adopted when the (satellite) capacity is precious [5-6]. The voice virtual queue is guaranteed to have at least $\frac{T_1}{T_1+T_2}$ of the total capacity and the data virtual queue $\frac{T_2}{T_1+T_2}$ of the total capacity.

There are two possible dequeue procedures for the rate-based control.

The first dequeue procedure of the rate-based control is implemented using a signal mechanism. The feasibility of this scheme depends on the frequency of signals. The signal is generated using a counter. The local CPU writes the count value to a ping-pong register. When the count value reaches the count, a signal occurs. When a signal occurs, the rate-based control sends out one packet in the queue. If the frequency of signals is beyond the capacity of the CPU, then instead of sending a packet, a block of packets can be sent . The line interface is to maintain the uplink synchronization. The line interface has both enqueue and dequeue schemes. The enqueue procedure is to insert the arrival packets from eight different rate-based control at the end of the queue. The dequeue procedure is implemented using a counter. The address generated by the counter reads out the packets in the queue. If no packet is available, idle packet is inserted in the stream.

The second dequeue procedure for the rate-based control is to use the departure sequence number (DSN) [5-8]. Each arrival packet is assigned a DSN. The assigned DSN is the transmission time for the packet at the rate-based control. The DSN assignment is performed on each rate-based control. Let us focus on the rate-based control 0. Assume the uplink rate is 8 packets per sec and the initial rate for rate-based control 0 is 1 packet per sec. The first arrival packet to rate-based control 0 is assigned the DSN_0. DSN_0 is equivalent to the current time. The time unit uses uplink packet transmission slot time. The uplink packet transmission slot time is 0.125 sec. The advantage of using the uplink transmission slot time as the time unit is that the number can be represented by an integer, which can be implemented by a counter. Assume the first packet arrival time is 0.25 sec. Then the DSN_0 is 2. The subsequent packets will be assigned a DSN according to the following equation:

$$DSN_{i+1} = \max \{\text{real time}, DSN_i + \text{information packet transmission slot time}\}.$$

In this example, the information transmission slot time is 8 times the uplink packet transmission slot time.

Assume the second packet arrives at the rate-based control 0 at uplink transmission slot time 5. Then DSN_1 is max {5, (2+8)}, which is 10. Assume the third packet arrives at the rate-based control 0 at uplink transmission time 19. Then DSN_1 is max {19, (10+8)}, which is 19. A send queue, where one buffer space corresponds to one packet, is employed. The enqueue procedure of the rate-based control is to insert the packet to the

corresponding buffer location according to the DSN. For example, a packet with DSN 2 is inserted at buffer location 2. If priority control is required, the DSN assignment has to consider the priority (see Figure 5-24). The priority of the packet is combined with the DSN into a new DSN. The new DSN is still the transmission time for the packet at the rate-based control. For example, the new DSN of a voice packet with a DSN 2 is 2. The new DSN of a data packet with a DSN 2 is 128+2. The new DSN will be reset at a fixed interval such that the buffer space can be reused. The new DSN is bounded by a threshold value. If the new DSN exceeds the threshold value, the packet is dropped. For example, assume the voice DSN is bounded by 128. If the voice DSN exceeds 128, the voice packet is dropped. The dequeue procedure is to read the packet in the queue based on the location of the packet in the queue. The enqueue procedure of the line interface stores the packets in a single queue. The dequeue procedure of the line interface sends out the packet in the queue using the uplink rate.

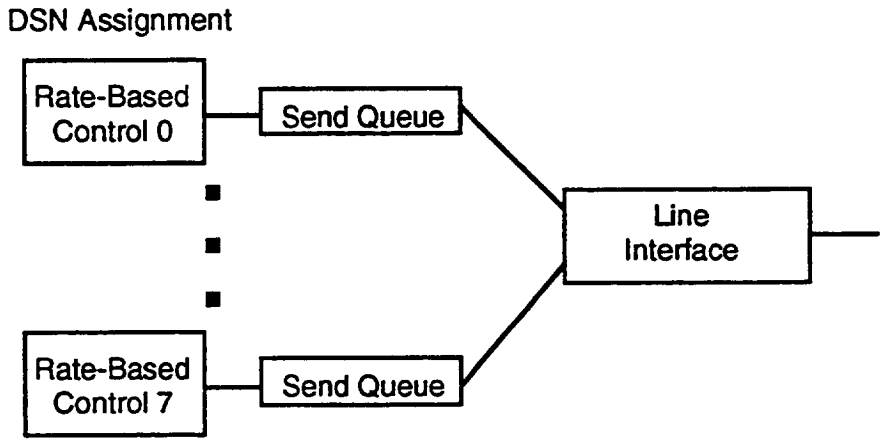

Figure 5-24. Send Queue is Associated with Each Rate-Based Control

The alternate implementation is to have only one send queue for different rate-based control (see Figure 5-25). The rate-based control assigns a DSN for each packet. The DSN and the priority of the packet is combined into a new DSN. The new DSN is used as the packet transmission time. The packets are stored in the queue based on the transmission time. The new DSN value will be reset at a fixed interval such that the buffer space can be reused. If two packets have the same transmission time, one packet is shifted to the next buffer location. The TDMA synchronization reads out the packets in the queues using the uplink rate. If there is no packet in a buffer location, an idle packet is sent out.

The implementation of the rate-based control and line interface for file formats II, III, IV, V and VI is similar to that for file format I and will not be discussed here.

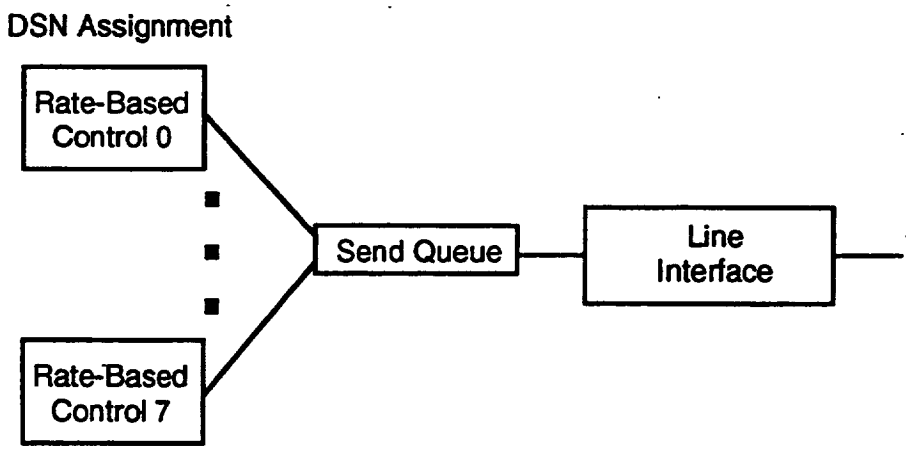

Figure 5-25. Send Queue is Shared by Eight Rate-Based Controls

If the information rate and the source rate can be reduced and increased at the same time, the functions of the rate-based control and the synchronization function of the line interface can be merged with the traffic source. The traffic source only has the dequeue procedure. Since the rate-based control is exercised on a per beam basis, the traffic source file is organized into eight different records, one record is for one downlink beam traffic (see Figure 5-26).

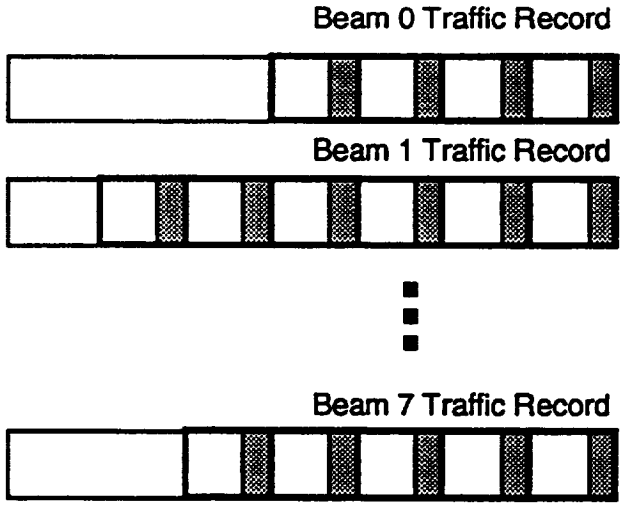

Figure 5-26. Modification of File Format to Accommodate Congestion Control when Source Rate and Information Rate can be Changed at the Same Time

The source response procedure to congestion for file format I, II and VI are discused first. File format I consists of idle and information packets. File format II consists of the idle and information packet headers. File format VI consists of the busy/idle status. The file is organized into 8 records, one record is for one downlink beam. File format I is used as the representative for the following discussion. The traffic source only has the dequeue procedure. A counter is used to read the corresponding packets at a record. When the network is congested, the packet are read out from the traffic file in a reused

rate and idle packets are inserted at fixed locations of the packet stream. In other words, the control action at the traffic generator when the network is congested is to enlarge the packet interarrival time. The configuration is shown in Figure 5-27.

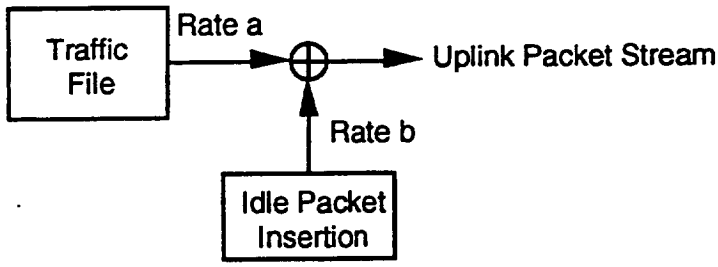

Figure 5-27. Traffic Generation Configuration B Considering Congestion Control

The counter reads (a x uplink transmission rate) pkt/sec from the traffic file and also inserts (b x uplink transmission rate) idle pkt/sec in the stream. Note there are both information packets and idle packets in the source file. The sum of a and b is equal to 1. The values of a and b are adaptable based on the satellite congestion status.

An example is given below. Initially, the TG generates traffic which results in switching loading of 0.9. No idle packets are inserted. In this case a = 1 and b = 0. However, there are 10% of idle packets in the traffic file. Then the the traffic loading for beam 1 is changed from 0.9 to 0.95. In this case, there are 5% of idle packets in the traffic file. To reduce the switch loading to 0.9, additional of 5.3 % of idle packets must be generated (i.e., b=0.053). The derivation for a and b is shown below.

$$\frac{0.95 \times a}{0.05 \times a + 0.95 \times a + b} = 0.9,$$

$$a + b = 1.0.$$

Therefore, a = 0.947 and b = 0.053.

These idle packets are inserted at fixed locations of the packet stream. The locations of the idle packets can be determined using table lookup and a counter. When the count matches a value in the lookup table, an idle packet (or a group of idle packets) is inserted at the packet stream. Different lookup tables are used for different values of b. These tables are pregenerated. Clearly, due to the generation of additional idle packets, the queueing delay of the information packets in the source file becomes longer. The queueing delay can be measured by computing the difference of arrival time stamp and the departure time stamp. Although this scheme has packet queueing capability, it has no priority control and packet dropping capability.

For file formats III, IV, and V, eight timers are used to generate signals to transmit information packets to the corresponding downlink beam. The value of a timer depends on the congestion status of the corresponding downlink beam. When the switch is congested, the timer value is increased. When a timer expires, the current time is compared with the time stamp of the next information packet header in the

corresponding record. If the current time is larger than the time stamp of the packet, the information packet is transmitted out. If the current time is smaller than the time stamp of the packet, it means the packet has not actually arrived yet; therefore, an idle packet should be generated. Although this scheme has packet queueing capability, it has no priority control and packet dropping capability.

For file format VII, the arrival times of the packet are not taken into consideration. Clearly, the arrival process of the packets can be assumed to be random. Since only a pool of sample information packet headers are available, to approximately represent the real traffic, idle packets are inserted at fixed locations of the output packet stream. For example, using an 8-bit counter, the number of idle packets which can be inserted in 255 packets is from 0 to 255. The number of idle packets, which should be generated, is a design decision. If there is no idle packet, the link utilization is 100% and the network is congested quickly. When the network is congested, then the number of idle packets which needs to be inserted is increased. To maintain a link utilization of 0.9, the number of idle packets which should be inserted should be about 26 packets. Inserting more idle packets is equivalent to reducing the information transmission rate. This scheme is the most unrealistic, but is has the simplest hardware. Although this scheme has packet queueing capability, it has no priority control and packet dropping capability.

5.4 Recommendation

Depending on the consideration factors, the recommendation for implementing the traffic generator will be different. A general comparison between the on-line and off-line approaches is provided in Table 5-9.

Table 5-9. Comparison between On-line and Off-line Traffic Generation

	storage	complexity	preprogramming time	selectable traffic patterns	cost
on-line	low	high	small	few	high
off-line	high	low	large	all	low

The first option is to use off-line dynamic traffic generation. Although the off-line approach is software-intensive, it has the advantages of low cost and flexibility of using any source queueing model. Two file formats are desirable. File format II requires the least amount of hardware; the packet arrival time and its associated header are pregenerated for the entire testing duration. File format VI has the least amount of memory requirement. The busy/idle status is pregenerated for the entire testing duration and only a small amount of sample packet headers are pregenerated. To select a packet from the sample packets, a random number is generated and inverse transform technique is applied to select a unique sample packet. Congestion control is achieved by inserting idle packets in the packet stream such that the packet interarrival time of the traffic file can be enlarged.

The second option is to user on-line dynamic traffic generation. Although the on-line approach is hardware-intensive, it has the advantages of little programming time and flexibility of changing the traffic characteristics in real time. The hardware to generate the packets can be either a workstation plus the host-network interface or dedicated hardware board. Congestion control is achieved by computing a new set of parameter values for the packet arrival process in real time.

5.5 Other Implementation Considerations

5.5.1 Transmission Link Emulation

To measure the end-to-end QOS, the impairment due to the satellite links needs to be taken into consideration. The satellite links generate two types of distortions: bit error and delay. The links generate random bit errors and bursty errors (if convolutional coding is used). The links also generate a constant propagation delay (from the terminal to the on-board switch) of about 0.135 sec.

There are two alternatives of emulating the transmission links (TLs). The first is to build a real TL circuit. The circuit is interfaced with the TG and the switch. The TL circuit introduces bit errors and delay to the arriving packets. This alternative is suited for on-line real time traffic generation. A simple way of introducing bit errors is to toggle a bit in the packet every N cycles.

An on-line pseudo packet transmission link module, which introduces loss and delay, for an ATM switch testbed was described in Reference 5-9. The pseudo packet transmission link module was used to emulate the transmission link characteristics. This pseudo transmission link module not only can introduce bit errors in packet or packet losses, it but also can introduce constant and variable delay for different connections. This link module can be used to emulate the satellite links and perceive the impairments of the links to the performance of real traffic such workstation traffic and LAN traffic. Currently, ADTECH SX/13 data channel simulator, which operates at 51.84 Mbit/sec, is available in the market.

The second is to introduce bit errors off-line. The network simulator can be programmed to generate bit errors based on the selected error distribution. Another (virtual) testing field is used in the packet header to indicate the number of bits is errored and the bit positions. Due to the error bits in the packet header, the packets may be routed to the wrong destination or may be discarded at the destination. To introduce a fixed end-to-end delay is quite simple. The TG will not generate traffic for 0.135 sec after the testbed starts running.

5.5.2 Feedback Traffic Generation

To perform congestion control, the network controller must generate congestion control messages on-line and in real time. However, since the frequency of sending congestion control messages is very infrequent, the hardware complexity is low. A pseudo delay element of 0.135 sec is required to insert between the TG and the network controller. In

general, the delay element is implemented by a RAM. The pseudo delay element can be a separate unit or be part of TG or network controller. Since the network controller is responsible of generating congestion control messages, it is easier for the network controller to introduce the delay. The network controller holds the message for 0.135 sec before it sends out the message.

The network controller broadcasts the congestion control message to all the output ports. There are two alternatives. The first is the OBSC sends the message through the local bus to the output port. The output port is programmed in such a way that the message is always inserted at a fixed location in a multiframe structure. The second alternative is that the OBSC participates output contention to send the message to all the output ports. The measurement results can be stored in registers of the output port. The OBSC can use a local bus to access the register information in individual output ports.

5.5.3 Packet Receiver Measurements

To analyze the performance of the switch, the characteristics of the traffic sources, and the effectiveness of the congestion control algorithms, different measurements have to be performed at the switch and the packet receiver. The measurements include

a) packet interarrival time

b) number of packets in a given interval

c) index of dispersion for counts

d) index of dispersion for intervals

e) number of loss packets for each priority

f) number of misinserted packets for each priority

g) total number of packets for each priority

h) packet switching delay (jitter) for each priority

i) packet end-to-end delay (jitter) for each priority

j) congestion control detection time

k) congestion control response time

l) congestion control recovery time

The packet interarrival time can be used to calculate the first two moments of the packet arrival process and used in the time comparison fitting algorithm discussed in Section 3. The number of packets in a given interval is required in the number

comparison fitting algorithm. The IDC and IDI can be used to characterize the packet arrival process. The number of loss packets and the total number of packets can be used to calculate the packet loss ratio. The distinction for packet loss due to errors, congestion at the satellite, or rate-based control at the traffic generator should be made. The total number of packets can be used to calculate the switch throughput. The packet switching delay can be used compute the mean average transfer delay and delay jitter.

The output traffic profile can be used to analyze the relationship between the output traffic pattern and the input traffic characteristics. The queue length should be measured at the output queue.

5.6 Commercially Available Traffic Generator

A survey of the available traffic generators is presented.

5.6.1 Microwave Logic PacketBERT-200

The PacketBERT-200 is a pattern generator with error injection capability and a pattern receiver. The generator can be operated from DC to 200 Mbit/sec. The data field (the packet payload) and the overhead field (the packet header) are user programmable. The data field can also be pseudo random bit sequence. Two separate RAMs are used to store the (prespecified) data field and the overhead field. The RAM size is 64 kbits. However, the sequence of reading the data from the RAM is deterministic. Although it is equipped with IEEE 488 and RS-232 bus, the bus is used only for off-line data download.

To be more flexible, a ping-pong data RAM and overhead RAM can be provided. The contents can be downloaded using the IEEE 488 or RS-232 bus such that different patterns can be generated rather than repeating the same pattern.

5.6.2 ADTECH ATM Cell Data Generator

The ATM cell data generator can generate up to 16 independent links. Each link may consist of multiple VPIs and VCIs. The arrival queueing model for one channel can be selected from one of the following models:

1. single manual: using keyboard to trigger a single cell.
2. burst manual: using keyboard to trigger a single burst of cells
3. single periodic: single cell arrivals with a fixed period from 2 to 2^{24} cells.
4. burst periodic: cell bursts with a fixed period from 2 to 2^{24} cells and a burst length of 1 to 2^{16} cells.

5. single Poisson: single cells arrivals following the Poisson distribution.

6. burst Poisson: arrivals of bursts following the Poisson distribution and a burst length of 1 to 2^{16} cells.

Each output link is a multiplexed stream of up to 7 different channels. The cell generator can hold up to 112 user-programmable cells (header and payload) and HEC errors. Each channel can select one of the 112 cells.

A cell generator with SONET interface, four output links and each running at 155.62 Mbit/sec is priced for $41,700.

5.6.3 HP 75000 Series 90 ATM Analyzer

HP announces a complete set of testing equipments for ATM. They can be used to test the physical layer, the ATM layer, and the AAL layer.

The HP E1691A ATM Generator generates cells with a rate of 149.76 Mbit/sec. The Generator can generate up to nine virtual channels. The cell header and payload are user programmable. The cell arrival distribution for each channel can select one of the following models:

1. single.

2. periodic.

3. single burst: a burst size of 1 to 8 cells.

4. periodic burst: a burst size of 1 to 8 cells.

The maximum payload size for AAL type 1 is 8 cells and for AAL type 2 is 16 cells. The Generator has error injection capability.

The ADTECH and HP traffic generators are very specific to ATM applications. Although the Microwave Logic traffic generator can generate packets with any size, it does not have the capability of generating traffic pattern following a certain statistical distribution.

Based on the high-level functional specifications for the TG, the configuration of the FPS test bed along with development support modules is shown in Figure 5-28. The functional requirements for each module are described below.

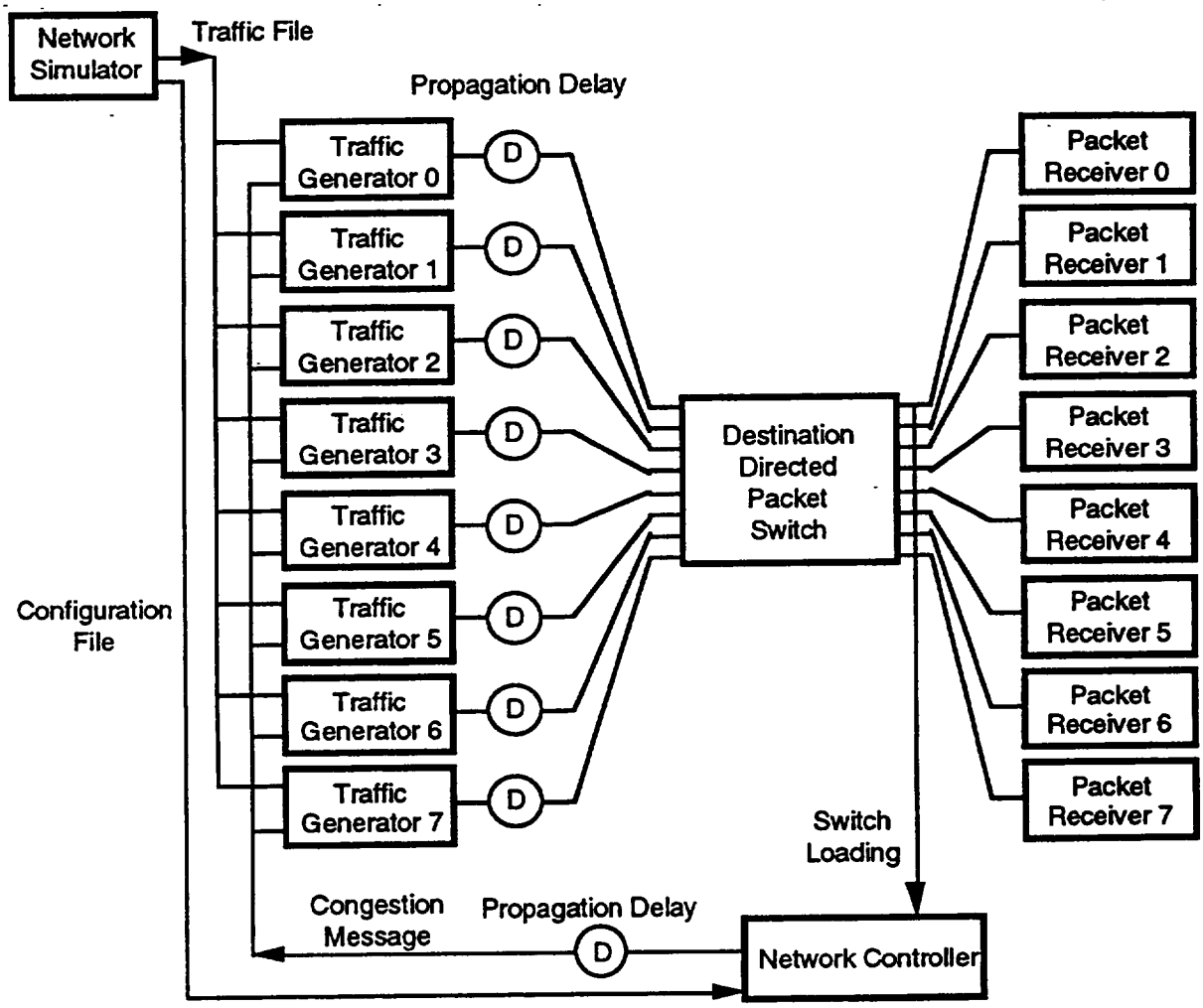

Figure 5-28. Functional Block Diagram of Testbed

5.7 Testbed Configuration

5.7.1 Network Simulator

The network simulator is the user interface to the testbed. The network simulator is responsible for traffic file generation and testbed configuration file generation. Both of the traffic file and configuration file can be saved for later use. It implements the generic traffic generator algorithm. Users choose the proper source queueing models for each TG or they can manually create the traffic file. Users also configure the testbed and define the performance measurement parameters. Users may want to collect all the measurements or select only a subset of the measurements. The testing duration can be either manually controlled (such as start and stop) or programmed. After the required parameter values are entered, the program will be running for a specified duration. The output of the program will contain traffic profiles for eight TGs. The duration of a traffic profile must be larger than the test duration for performance

measurement. Then the network simulator downloads the traffic profiles to the respective TG.

5.7.2 Traffic Generator

The TG takes the traffic profile from the network simulator as input. All eight TGs are synchronized so that they can start generating packets at the same instant. Then the TG generates electronic packets in real time based on the traffic profile.

The TG will respond to the congestion message received from the network controller. When the switch is congested, the TG will execute the traffic reduction algorithm. When the switch is underutilized, the TG will execute the traffic increase algorithm.

5.7.3 Network Controller

The function of the network controller is to communicate with different modules in the testbed and verify the testbed operation. At the beginning of the test, it receives the configuration file from the network simulator and performs hardware initialization and parameter setting. For example, it accepts the input parameters for the congestion detection algorithm. It synchronizes all the modules in the testbed. It initializes the packet receiver so that the measurement data can be collected. At the end of testing, the measurement data will be downloaded to the network simulator for performance analysis. When it receives the switch loading from the switch, it performs congestion detection algorithm. The algorithm translates the switching loading into a congestion message, which contains the traffic reduction/increase information. It is desirable if the network controller can display the measurement data on line and in real time.

5.7.4 Packet Receiver

The function of the packet receiver is to ack as a packet sink, perform measurements and store data. The measurement should be performed for each VCN. The performance measurements include switch loading,number of lost packets, bit errors on packets, packet delay, packet delay jitter and misinserted packets. The arrival packets are stored in an output file for postprocessing. It is desirable the users can specify the number of packets to be captured for a particular VCN or for the entire packet stream. The start of capture can be triggered by the user manually or by an event defined by the users. The possible events are the number of errored packets exceeds a limit or the delay jitter exceeding a limit.

Section 6

Conclusion

There are several traffic models that have been widely used in the past to test telecommunications equipment and terrestrial/satellite systems. A difficulty arises in adopting a particular model to emulate various types of live traffic. Traffic types radically vary according to telecommunications services (voice, stream data, interactive data, low/high speed video, image), user communities (service industry, manufacturing, banking, hospitals, etc.), and communications media (e.g. terrestrial, fixed satellites, and mobile satellites). Even within a very small aperture terminal (VSAT) network, traffic characteristics vary from one service application to another. Therefore, flexibility to generate different types of simulated traffic becomes most important in designing a traffic generator.

Among the several alternatives investigated in this report, the Markov-Modulated Poisson Process (MMPP) is one of the most flexible models and is recommended for implementation. This model can generate a mix of circuit and packet switched traffic and has also been widely accepted as a better model for emulating real traffic. With proper selection of parameters, the MMPP degenerates to a Poisson process as a special case. Although the MMPP model provides flexible traffic emulation, the traffic generator should be designed to allow accommodation of other models to emulate user/service specific traffic scenarios.

Congestion is an inherent characteristic of a packet switched system and requires a proper control mechanism to avoid performance degradation. A tradeoff exists between the size of on-board buffer and the switch performance (i.e., a throughput, packet loss ratio, delay, and delay jitters). Although analytical as well as simulation results have been presented in the previous studies (Task Order #2 and SCAR BISDN study), this is a complex issue and will be most effectively handled using a testbed facility. Real time operation with flexible traffic models will emulate the operational system and will provide more realistic results on the effectiveness of a selected control algorithm. The report includes design guidelines for selecting proper traffic simulator parameters to perform congestion control experiments.

A traffic generator may be implemented using one of two approaches. The first approach uses hardware to generate packet arrival times according to a selected traffic model. The second design approach utilizes a separate workstation to generate a traffic profile off-line (arrival times) according and implements packet multiplexing and transmission functions in real time in the traffic generator. The hardware-based system is more complex but allows real-time operation of a traffic generator to emulate various traffic flow scenarios. The software-based system is more flexible and possibly lower cost to develop. However, traffic profile generation for different traffic patterns may take significantly more time than the hardware-based system (in the order of several hours).

In response to a congestion control message, the traffic generator dynamically controls its output by increasing or decreasing the inter-packet transmission time. The network controller performs processing of switch congestion status messages, implements a congestion control algorithm, and generates control messages for the traffic generators. In the operational system, these functions may be incorporated into the user terminals.

Section 7

References

[1-1] W. Ivancic et al, "Destination Directed Packet Switch Architecture for a Geostationary Communication Satellite Network," World Space Congress, 1992.

[1-2] J. Shoch and J. Hupp, "Measured Performance of an Enternet Local Network," Communications of the ACM, vol. 23, no. 12, Dec. 1980.

[3-1] K. Khalil, "A Real-time Algorithm for Burstiness Analysis of Network Traffic," ICC, pp. 521-527, 1992.

[3-2] R. Jain and S. Routhier, "Packet Trains - Measurements and a New Model for Computer Network Traffic," IEEE JSAC, vol. 4, no. 6, pp. 986-995, Sep. 1986.

[3-3] R. Gusella, " Characterizing the Variability of Arrival Processes with Indexes of Dispersion," IEEE JSAC, vol. 9, no. 2, Feb. 1991.

[3-4] H. Yamada and S. Sumita, "A Traffic Measurement Method and its Application for Cell Loss Probability Estimation in ATM Networks," IEEE JSAC, vol. 9, no. 3, pp. 315-323, April 1991.

[3-5] J. Bae and T. Suda, "Survey of Traffic Control Schemes and Protocols in ATM Networks," Proceedings of the IEEE, vol. 79, no. 2, pp. 170-189, Feb. 1991.

[3-6] G. Woodruff and et al, "Control of ATM Statistical Multiplexing Performance," Computer Networks and ISDN Systems, pp. 351-360, 1990.

[3-7] V. Ramaswami and G. Latouche, "Modeling Packet Arrivals From Asynchronous Input Lines," ITC 12, pp. 721-727, 1989.

[3-8] T. Eliazov, and et al, "Performance of an ATM Switch: Simulation Study," INFOCOM, pp. 644-659, 1990.

[3-9] A. Descloux, "Stochastic Models for ATM Switching Networks," IEEE JSAC, vol. 9, no. 3, pp. 450-457, April 1991.

[3-10] S. Liew, "Performance of Input-Buffered and Output-Buffered ATM Switches under Bursty Traffic: Simulation Study," GLOBEOCM, pp. 1919-1925, 1990.

[3-11] J. Hui, Switching and Traffic Theory for Integrated Broadband Networks, Massachusetts: Kluwer Academic Publishers, 1990.

[3-12] S. Kowtha and D. Vaman, "A Generalized ATM Traffic Model and its Application in Bandwidth Allocation," ICC, pp. 1009-1013, 1992.

[3-13] H. Heffes and D. Lucantoni, "A Markov Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," IEEE JSAC, pp. 856-867, 1986.

[3-14] A. Baiocchi and et al, "Loss Performance Analysis of an ATM Multiplexer Loaded with High-Speed On-Off Sources," IEEE JSAC, vol. 9, no. 3, pp. 388-393, 1991.

[3-15] S. Falaki and S.-A. Sorensen, "Traffic Measurements on a Local Area Computer Network," Computer Communications, vol. 15, no. 3, pp. 193-197, April 1992.

[3-16] "Explicit Forward Congestion Notification in ATM," T1S1 91-585, Dec. 1991.

[3-17] J. Daigle and J. Langford, "Models for Analysis of Packet Voice Communications Systems," IEEE JSAC, vol. 4, no. 6, pp. 847-855, Sep. 1986.

[3-18] M. Hirano and N. Watanable, "Characteristics of a Cell Multiplexer for Bursty ATM Traffic," ICC, pp. 399-403, 1990.

[3-19] T. Kamitake and T. Suda, "Evaluation of an Admission Control Scheme for an ATM Network Considering Fluctuations in Cell Loss Rate," Globecom, pp. 1774-1780, 1989.

[3-20] D.-J. Shyy, "Network Control System Protocol Simulation Study," Final Report, Prepared for AMSC/TMI, May 1991.

[3-21] M. Hluchyj and A. Bhargava, "Queueing Disciplines for Integrated Fast Packet Networks," ICC, pp. 990-996, 1992.

[3-22] L. Kleinrock, Queueing Systems Vol I: Theory, New York: Wiley, 1974.

[3-23] J.-Y. Le Boudec, "An Efficient Solution for Markov Models of ATM Links with Loss Priorities," IEEE JSAC, vol. 9, no. 3, pp. 408-417, April 1992.

[3-24] K. Sriram and W. Whitt, "Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data," IEEE JSAC, vol. 4, no. 6, pp. 833-846, Sep. 1986.

[3-25] M. Murata and et al, "Analysis of a Discrete-Time Single-Server Queue with Bursty Inputs for Traffic Control in ATM Networks," IEEE JSAC, vol. 8, no. 3, pp. 447-458, April 1990.

[3-26] I. Norros, and et al, "The Superposition of Variable Bit Rate Sources in an ATM Multiplexer," IEEE JSAC, vol. 9, no. 3, pp. 378-387, April 1991.

[3-27] E. Rathgeb, "Modeling and Performance Comparison of Policing Mechanims for ATM Networks," IEEE JSAC, vol. 9, no. 3, pp. 325-334, April 1991.

[3-28] F. Guillemin and A. Duplis, "A Basic Requirement for the Policing Function in ATM Networks," Computer Networks and ISDN Systems, vol. 24, pp. 311-320, May 1992.

[3-29] T. Yang and D. Tsang, "A Novel Approach to Estimating the Cell Loss Probability of an ATM Multiplexer Loaded with Homogeneous Bursty Sources," pp. 511-517, 1992.

[3-30] K. Meier-Hellstern, "The Analysis of a Queue Arising in Overflow Models," IEEE Transactions on Communications, vol. 37, no. 4, pp. 367-372, 1989.

[3-31] W. Press and et. al., Numerical Recipes in C, Cambridge University Press: New York, 1988.

[4-1] "Modeling and Simulation of Congestion Control in a Destinaton Directed Packet Switch for Satellite Communications", Final Report, NASA Contrace NAS3-25933, Prepared by COMSAT Laboratories, Nov. 1992.

[5-1] O. Aboul-Magd and M. Wernik, "Traffic Experimentation in ATM Testbed", Globecom, pp. 1445-1449, 1990.

[5-2] J. Banks and J. Carson, II, Discrete-Event System Simulation, Prentice-Hall Inc, New Jersey, 1984.

[5-3] B. Wichmann and I. Hill, "An Efficient and Portable Pseudo-random Number Generator," Appl. Stat., Algorithm AS 183, pp. 188-190, 1982.

[5-4] A. Lazar, G. Pacifici, and J. White, "Real-Time Traffic Measurements on MAGNET II," IEEE JSAC, vol. 8, no. 3, pp. 467-483, April 1990.

[5-5] O. Aboul-Magd and H. Gilbert, "Incorporating Congestion Feedback in B-ISDN Traffic Management Strategy," ISS, pp. 12-16, 1992.

[5-6] H. Zhang adnd S. Keshav, "Comparison of Rate-Based Service Discipline," SIGCOMM, pp. 113-121, 1991.

[5-7] K. Sriram, "Dynamic Bandwidth Allocation and Congestion Control Schemes for Voice and Data Multiplexing in Wideband Packet Technology," ICC, pp. 1003-1008, 1990.

[5-8] H. J. Chao, "A Novel Architecture for Queue Management in the ATM Network," IEEE JSAC, vol. 9, no. 7, pp. 1110-1118, Sep. 1991.

[5-9] K. Yamazaki and et al, "Flexible QOS Communications for Integrated Video Service in B-ISDN," ISS, pp. 311-315, 1992.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1996	Final Contractor Report

4. TITLE AND SUBTITLE

Data Generation for Destination Directed Packet Switch

5. FUNDING NUMBERS

WU-235-01-04
C-NAS3-25933

6. AUTHOR(S)

D.J. Shyy and Tom Inukai

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Comsat Laboratories
22300 Comsat Drive
Clarksburg, Maryland 20871

8. PERFORMING ORGANIZATION REPORT NUMBER

E-8361

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR-194452

11. SUPPLEMENTARY NOTES

Project Manager, Jorge A. Quintana, Space Electronics Division, NASA Lewis Research Center, organization code 5650, (216) 433-6519.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 17

This publication is available from the NASA Center for AeroSpace Information, (301) 621-0390.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

A Destination Directed Packet Switch (DDPS) for future advanced satellite communications employs a routing concept based on the routing information contained in individual packet headers. Several institutions, including NASA's Lewis Research Center, are currently developing a proof-of-concept DDPS to exploit its capabilities for providing future multimedia user services. One critical issue for a DDPS architecture is the development of testing procedures under various simulated traffic environments. Although some commercial test equipments are currently available for testing general purpose switching systems and Asynchronous Transfer Model (ATM) switches, they are often constrained by simple traffic models, fixed packet formats and no dynamic control of traffic sources. The purpose of this study is to investigate a design concept of a traffic generator for testing an on-board DDPS in a testbed environment.

14. SUBJECT TERMS

Traffic generation; Congestion control; Packet switching

15. NUMBER OF PAGES

139

16. PRICE CODE

A07

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

National Aeronautics and
Space Administration

Lewis Research Center
21000 Brookpark Rd.
Cleveland, OH 44135-3191